FRIEDRICH-SCHILLER-
UNIVERSITÄT
JENA

Development of deep learning applications for the automated extraction of chemical information from scientific literature

Dissertation
(kumulativ)

zur Erlangung des akademischen Grades doctor rerum naturalium
(Dr. rer. nat)

vorgelegt dem Rat der Chemisch-Geowissenschaftlichen Fakultät der
Friedrich-Schiller-Universtität Jena

von M. Sc. Henning Otto Brinkhaus
geboren am 20.10.1993 in Lingen (Ems)

Gutachter:

1. Gutachter:	Prof. Dr. Christoph Steinbeck, Friedrich-Schiller-Universität, Jena
2. Gutachter:	Prof. Dr. Achim Zielesny, Westfälische Hochschule, Recklinghausen
3. Gutachter:	Prof. Dr. Matthias Rarey, Universität Hamburg, Hamburg
Tag der Verteidigung:	11. Oktober 2023

For my family.

If I have seen further,
it is by standing
on the shoulders of giants.

ISAAC NEWTON

Table of contents

List of Figures

List of Tables

List of Abbreviations

ABC-Net	Atom and Bond Center Network.
AI	Artificial Intelligence.
BERT	Bidirectional Encoder Representations from Transformers.
CDDD	Continuous and Data-Driven Molecular Descriptor.
CDK	Chemistry Development Kit.
CLiDE	Chemical Literature Data Extraction.
CNN	Convolutional Neural Network.
DBSCAN	Density-Based Spatial Clustering of Application with Noise.
DECIMER	Deep lEarning for Chemical IMagE Recognition.
ECFP	Extended Connectivity Fingerprints.
EIDD	Emory Institute for Drug Development.
FAIR	Findable, Accessible, Interoperable, Reusable.
GRU	Gated Recurrent Units.
InChI	IUPAC International Chemical Identifier.
IUPAC	International Union of Pure and Applied Chemistry.
LLM	Large Language Model.
LSTM	Long Short-Term Memory.
MACCS	Molecular ACCess System.
Mask R-CNN	Mask Region-based Convolutional Neural Network.
MICER	Molecular Image Captioner.
MSE-DUDL	Molecular Structure Extraction from Documents Using Deep Learning.
NER	Named Entity Recognition.
NFDI	Nationale ForschungsDatenInfrastruktur.
NLP	Natural Language Processing.

NMR	Nuclear Magnetic Resonance.
OCR	Optical Character Recognition.
OCSR	Optical Chemical Structure Recognition.
ODOSOS	Open Data, Open Source, Open Standards.
OOD	Out-Of-Distribution.
ORD	Open Reaction Database.
OSRA	Optical Structure Recognition Application.
PDF	Portable Document Format.
PIKAChU	Python-based Informatics Kit for Analysing Chemical Units.
QSAR	Quantitative Structure-Activity Relationship.
QSPR	Quantitative Structure-Property Relationship.
RNN	Recurrent Neural Network.
SELFIES	SELF-referencIng Embedded Strings.
SMILES	Simplified Molecular Input Line Entry System.
STOUT	SMILES to IUPAC Translator.
USPTO	United States Patent and Trademark Office.

Summary

In scientific literature, chemical information is typically published in text and images intended to be read by humans. As natural language and images are unstructured data formats, machines cannot directly interpret the information included in them. The availability of open repositories for machine-readable data is relatively rare in chemistry. Instead, the scientific literature is often the only way chemical knowledge is published. Recent advancements in deep learning have yielded impressive results, but these data-driven technologies depend on the open availability of data. As long as digital research data management systems and open data repositories are not yet widely adopted in the chemical research community, the sole approach to handle the absence of structured chemical data is to extract it from the literature. The manual extraction of chemical information from the literature is an error-prone and time-consuming process. Moreover, it is an expensive procedure because it requires highly skilled workers with expertise in the field to decide what information is relevant to extract. Consequently, methods for the automated extraction of chemical information from the scientific literature are desirable to make chemical knowledge accessible in a simple and cost-effective way.

This thesis deals with the development of deep learning applications for the automated extraction of chemical information from the scientific literature. Here, the main objective is to extract molecular structures from images. Extracting molecular structures from the scientific literature requires multiple steps. First, chemical structures must be detected and segmented in a given document from whole pages. Then, the segmented chemical structure depictions must be translated into machine-readable representations of the depicted molecules, which is referred to as Optical Chemical Structure Recognition (OCSR).

Until today, DECIMER Segmentation, which has been developed during the work on this thesis, is the only deep-learning-based open-source application for the segmentation of chemical structures from the scientific literature. It relies on the Mask Region-based Convolutional Neural Network (Mask R-CNN) architecture. The model has been trained based on 9992 manually annotated regions that contain chemical structure depictions on 1820 pages from publications from the Journal of Natural Products. The Mask R-CNN model returns one mask per instance of a chemical structure depiction. A mask is a matrix that describes the region that contains the chemical structure depiction in the original image. The masks are then refined and completed using a mask expansion algorithm. DECIMER Segmentation can process complete Portable Document Format (PDF) documents. All pages are converted to images, the Mask R-CNN model generates the original masks, and the mask expansion algorithm then generates the refined, complete masks. Finally, the chemical structure depictions are segmented based on the final masks, resulting in separate images containing only chemical structure depictions.

The OCSR application DECIMER Image Transformer has been further developed as a part

of the work presented herein. The current version uses an encoder-decoder architecture based on EfficientNetV2-M and a transformer to translate chemical structure depictions into the SMILES representation of the depicted molecules. The model has been trained on more than 450 million pairs of chemical structure depictions and the corresponding SMILES representations of the depicted structures. It can interpret a wide variety of depiction styles and even some hand-drawn structures. It can interpret various functional group and superatom labels as well as common R-group placeholder variables in depictions of Markush structures. In the comparative performance analysis presented herein, DECIMER Image Transformer yields very competitive results. Altogether, it represents a reliable solution to the problem of translating chemical structure depictions into machine-readable representations.

DECIMER Image Transformer is an entirely data-driven deep-learning-based OCSR application which relies on a supervised learning process. As it learns to process images during training, the diversification of the training data is an essential factor for the tool's capability to generalise well on all kinds of chemical structure depictions. Besides covering a broad range of chemical space, it is essential to represent various ways of depicting chemical structures in the data so that the resulting model can learn to interpret them independently of the specific depiction style. Only then the trained model can process the variety of chemical structure depictions found in the scientific literature. A comprehensive artificial OCSR training data generation and diversification strategy has been implemented in the form of the application RanDepict. It ensures diverse chemical structure depictions by pseudo-randomly scrambling all available depiction functionalities of the cheminformatics toolkits Chemistry Development Kit (CDK), RDKit, Indigo and the Python-based Informatics Kit for Analysing Chemical Units (PIKAChU). RanDepict can also guarantee the diverse coverage of the depiction feature space in a set of chemical structure depictions by using the MaxMin algorithm to pick diverse sets of depiction features internally represented as binary vectors. This way, RanDepict can generate diverse datasets of chemical structure depictions that can be used for training deep-learning-based OCSR models, which are robust and can generalise effectively.

The automated interpretation of hand-drawn chemical structures is a challenging task due to the variety of personal drawing styles and the lack of normalisation. Although training data for the DECIMER Image Transformer did not include any hand-drawn structures, during the development, it became apparent that the application could interpret some of these structure depictions. A benchmark dataset was needed to systematically evaluate the tool's ability to interpret structure depictions of this kind. Consequently, the DECIMER hand-drawn molecule image dataset was created. The dataset is a set of 5088 manually drawn chemical structure depictions published along with machine-readable representations of the depicted molecules. The images were manually drawn by 24 volunteers from the University of Applied Sciences in Recklinghausen, Germany.

DECIMER Segmentation and DECIMER Image Transformer were integrated into the DECIMER.ai web application. DECIMER.ai offers a user interface that enables users without programming knowledge to extract chemical information from the scientific literature. After uploading a PDF document, DECIMER Segmentation is used to detect and segment all chemical structure depictions. After the subsequent processing of the images with the DECIMER Image Transformer, the segments and their corresponding SMILES representations are displayed, and the structure is loaded into an embedded chemical structure editor. Here, they can be modified before downloading the segmented images and machine-readable representations of the depicted molecules. Alternatively, chemical structure depictions can be uploaded directly. In that case, the DECIMER Image Classifier, a binary classifier capable of distinguishing between chemical structure depictions and other images, produces a warning if the images are not chemical structure depictions. Then, the images are processed by the DECIMER Image Transformer and the structures are presented as described above. Until today, the DECIMER.ai web application is the only open-source application that combines the segmentation and interpretation of chemical structure depictions from the literature in a comprehensive user interface.

As the work presented herein aims to make chemical information publicly available in open databases, recent advances in molecular informatics enabled by combining data-driven deep-learning methods and openly available chemical data have been reviewed. AI-driven progress depends on the availability of data for training. In the areas where data was available in the past, data-driven technologies have yielded impressive results. Examples are the progress in synthesis planning using transformer models, new approaches to solve the protein folding problem or advancements in natural product-based drug discovery. The applications developed during the work on this thesis aim at making chemical information publicly accessible to enable similar progress in other areas of chemistry in the future.

The source code of applications developed during the work on this thesis is openly available under permissive licenses. All datasets have been published following the FAIR data standards, which means that they are findable, accessible, interoperable and reusable, and are publicly accessible. This way, researchers or other organisations can use and adapt the software and the datasets freely according to their needs. As the software's source code is publicly available, it has the potential to profit from contributions and suggestions from the user community. In future, the DECIMER.ai application can be extended to include the capability to process textual information. Additionally, the chemical literature mining software presented herein can be integrated into the submission pipelines of open databases where a human curator only needs to validate the automatically extracted information. The work presented herein is a contribution to the open availability of chemical data and has the potential to develop further in the future.

Zusammenfassung

In der wissenschaftlichen Literatur werden chemische Informationen in der Regel in Form von Texten und Bildern veröffentlicht, die darauf ausgelegt sind, von Menschen gelesen zu werden. Da natürliche Sprache und Bilder unstrukturierte Datenformate sind, können Maschinen die darin enthaltenen Informationen nicht direkt interpretieren. Die Verfügbarkeit offener Repositories für maschinenlesbare Daten ist in der Chemie relativ selten. Stattdessen stellt die wissenschaftliche Literatur oft die einzige Möglichkeit dar, chemisches Wissen zu veröffentlichen. Die jüngsten Fortschritte im Bereich des Deep Learning haben zu beeindruckenden Ergebnissen geführt, aber diese datengetriebenen Technologien hängen von der offenen Verfügbarkeit von Daten ab. Solange digitale Systeme zur Verwaltung von Forschungsdaten und offene Datenrepositorien in der chemischen Forschung noch nicht weit verbreitet sind, besteht der einzige Ansatz zur Bewältigung des Mangels an strukturierten chemischen Daten darin, sie aus der Literatur zu extrahieren. Die manuelle Extraktion von chemischen Informationen aus der Literatur ist ein fehleranfälliger und zeitaufwändiger Prozess. Es ist zudem ein kostspieliges Verfahren, da es hochqualifizierte Mitarbeiter mit Fachkenntnissen auf diesem Gebiet erfordert, um zu entscheiden, welche Informationen für die Extraktion relevant sind. Daher sind Methoden zur automatischen Extraktion von chemischen Informationen aus der wissenschaftlichen Literatur wünschenswert, um chemisches Wissen auf eine einfache und kostengünstige Weise zugänglich zu machen.

Diese Arbeit beschäftigt sich mit der Entwicklung von Deep-Learning-Anwendungen für die automatisierte Extraktion von chemischen Informationen aus der wissenschaftlichen Literatur. Das Hauptziel ist dabei die Extraktion molekularer Strukturen aus Bildern. Die Extraktion von Molekülstrukturen aus der wissenschaftlichen Literatur ist ein mehrschrittiger Prozess. Zunächst müssen die chemischen Strukturen in einem gegebenen Dokument erkannt und segmentiert werden. Anschließend müssen die segmentierten chemischen Strukturabbildungen in maschinenlesbare Darstellungen der abgebildeten Moleküle übersetzt werden, was als Optical Chemical Structure Recognition (OCSR) bezeichnet wird.

Bis heute ist DECIMER Segmentation, das während der Arbeit an dieser Dissertation entwickelt wurde, die einzige Deep-Learning-basierte Open-Source-Anwendung für die Segmentierung chemischer Strukturen aus der wissenschaftlichen Literatur. Es basiert auf der Mask Region-based Convolutional Neural Network (Mask R-CNN) Architektur. Das Modell wurde auf mithilfe von 9992 manuell annotierten Regionen trainiert, die Darstellungen chemischer Strukturen auf 1820 Seiten aus Veröffentlichungen des Journal of Natural Products enthalten. Das Mask R-CNN-Modell liefert eine Maske pro Instanz einer chemischen Strukturabbildung. Eine Maske ist eine Matrix, die die Region, die die chemischen Strukturabbildung im Originalbild einnimmt, beschreibt. Die Masken werden dann mit einem Maskenexpansionsalgorithmus verfeinert und vervollständigt. DECIMER Segmentation kann komplette Dokumente im Portable Document (PDF) Format verarbeiten. Alle

Seiten werden in Bilder umgewandelt, das Mask R-CNN-Modell erzeugt die ursprünglichen Masken, und der Maskenerweiterungsalgorithmus erzeugt dann die verfeinerten, vollständigen Masken. Schließlich werden die Darstellungen der chemischen Struktur auf der Grundlage der endgültigen Masken segmentiert, so dass separate Bilder entstehen, die nur die Darstellungen der chemischen Strukturen enthalten.

Die OCSR-Anwendung DECIMER Image Transformer wurde im Rahmen der hier vorgestellten Arbeit weiterentwickelt. Die aktuelle Version verwendet eine Encoder-Decoder-Architektur, die auf EfficientNetV2-M und einem Transformer basiert, um chemische Strukturabbildungen in die SMILES-Darstellung der abgebildeten Moleküle zu übersetzen. Das Modell wurde mit mehr als 450 Millionen Paaren von chemischen Strukturabbildungen und den entsprechenden SMILES-Repräsentationen der dargestellten Strukturen trainiert. Es kann eine breite Palette von Darstellungsstilen und sogar einige handgezeichnete Strukturen interpretieren. Es kann verschiedene Label für funktionelle Gruppen und strukturelle Elemente sowie gängige R-Gruppen-Platzhaltervariablen in Darstellungen von Markush-Strukturen interpretieren. In der hier vorgestellten vergleichenden Leistungsauswertung hat DECIMER Image Transformer sehr konkurrenzfähige Ergebnisse erzielt. Insgesamt stellt es eine zuverlässige Lösung für das Problem der Übersetzung von chemischen Strukturabbildungen in maschinenlesbare Darstellungen dar.

DECIMER Image Transformer ist eine vollständig datengetriebene Deep-Learning-basierte OCSR-Anwendung, die auf einem überwachten Lernprozess beruht. Da das Modell während des Trainings lernt, Bilder zu verarbeiten, ist die Diversifizierung der Trainingsdaten ein wesentlicher Faktor für die Fähigkeit des Tools, auf alle Arten von chemischen Strukturabbildungen anwendbar zu sein. Neben der Abdeckung eines breiten Spektrums des chemischen Raums ist es von entscheidender Bedeutung, verschiedene Arten der Darstellung chemischer Strukturen in den Daten zu repräsentieren, damit das resultierende Modell lernen kann, sie unabhängig vom spezifischen Darstellungsstil zu interpretieren. Nur dann kann das trainierte Modell die Vielfalt der Darstellungen chemischer Strukturen, wie sie in der wissenschaftlichen Literatur zu finden ist, verarbeiten. Mit der Anwendung RanDepict wurde eine umfassende Strategie zur künstlichen OCSR-Trainingsdatengenerierung und -diversifizierung implementiert. Sie erzeugt vielfältige Darstellungen chemischer Strukturen, indem sie alle Funktionalitäten zur Darstellung chemischer Strukturen der Chemoinformatik-Toolkits Chemistry Development Kit (CDK), RDKit, Indigo und des Python-based Informatics Kit for Analysing Chemical Units (PIKAChU) pseudo-zufällig kombiniert. RanDepict kann auch die diverse Abdeckung des Merkmalsraums in einem Satz von chemischen Strukturabbildungen garantieren, indem es den MaxMin-Algorithmus verwendet, um verschiedene Sätze von Darstellungsmerkmalen auszuwählen, die intern als binäre Vektoren dargestellt werden. Auf diese Weise kann RanDepict verschiedene Datensätze mit Darstellungen chemischer Strukturen generieren, die für das Training von Deep-Learning-basierten OCSR-Modellen verwendet werden können, die robust sind und effektiv generalisieren können.

Die automatisierte Interpretation handgezeichneter chemischer Strukturen ist aufgrund der Vielfalt der persönlichen Zeichenstile und der fehlenden Normierung eine anspruchsvolle Aufgabe. Obwohl die Trainingsdaten für den DECIMER Image Transformer keine handgezeichneten Strukturen enthielten, wurde während der Entwicklung deutlich, dass die Anwendung einige handgezeichnete chemische Strukturabbildungen interpretieren kann. Es wurde ein Benchmark-Datensatz benötigt, um die Fähigkeit des Tools zur Interpretation dieser Art von Strukturabbildungen systematisch zu bewerten. Daher wurde der DECIMER-Datensatz für handgezeichnete Moleküldarstellungen erstellt. Der Datensatz besteht aus 5088 manuell gezeichneten chemischen Strukturabbildungen, die zusammen mit maschinenlesbaren Darstellungen der abgebildeten Moleküle veröffentlicht wurden. Die Bilder wurden von 24 Freiwilligen der Westfälischen Hochschule in Recklinghausen, Deutschland, gezeichnet.

DECIMER Segmentation und DECIMER Image Transformer wurden in die Webanwendung DECIMER.ai integriert. DECIMER.ai bietet eine Benutzeroberfläche, die es Benutzern ohne Programmierkenntnisse ermöglicht, chemische Informationen aus der wissenschaftlichen Literatur zu extrahieren. Nach dem Hochladen eines PDF-Dokuments wird DECIMER Segmentation verwendet, um alle chemischen Strukturabbildungen zu erkennen und zu segmentieren. Nach der anschließenden Verarbeitung der Bilder mit dem DECIMER Image Transformer werden die Segmente und ihre entsprechenden SMILES-Darstellungen angezeigt, und die Struktur wird in einen eingebetteten Editor für chemische Strukturen geladen. Hier können sie bearbeitet werden, bevor die segmentierten Bilder und maschinenlesbaren Darstellungen der dargestellten Moleküle heruntergeladen werden. Alternativ können chemische Strukturabbildungen auch direkt hochgeladen werden. In diesem Fall wird der DECIMER Image Classifier, ein binärer Klassifikator, der in der Lage ist, zwischen chemischen Strukturabbildungen und anderen Bildern zu unterscheiden, eine Warnung aus, wenn es sich bei den Bildern nicht um chemische Strukturabbildungen handelt. Anschließend werden die Bilder mit dem DECIMER Image Transformer verarbeitet, und die Strukturen werden wie oben beschrieben dargestellt. Bis heute ist die Webanwendung DECIMER.ai die einzige Open-Source-Anwendung, die die Segmentierung und Interpretation von chemischen Strukturabbildungen aus der Literatur in einer umfassenden Benutzeroberfläche vereint.

Da die hier vorgestellte Arbeit darauf abzielt, chemische Informationen öffentlich zugänglich zu machen, wurden die jüngsten Fortschritte in der Molekularinformatik, die durch die Kombination von datengetriebenen Deep-Learning-Methoden und offen verfügbaren chemischen Daten ermöglicht wurden, in einem Review-Artikel zusammengefasst. Der datengegetriebene Fortschritt hängt von der Verfügbarkeit von Daten für das Training ab. In den Bereichen, in denen in der Vergangenheit Daten verfügbar waren, haben datengesteuerte Technologien beeindruckende Ergebnisse erzielt. Beispiele sind die Fortschritte bei der Planung von Synthesen mit Hilfe von Transformer-Modellen, neue Ansätze zur Lösung des Proteinfaltungsproblems oder Fortschritte bei der Entdeckung von Arzneimitteln auf der Grundlage von Naturstoffen. Die im Rahmen dieser Arbeit entwickelten Anwendungen zielen darauf ab, chemische Informationen öffentlich zugänglich zu machen, um in Zukunft ähnliche Fortschritte in anderen Bereichen der Chemie zu ermöglichen.

Der Quellcode der Anwendungen, die während der Arbeit an dieser Dissertation entwickelt wurden, ist unter freizügigen Lizenzen frei verfügbar. Alle Datensätze wurden gemäß den FAIR-Datenstandards veröffentlicht, was bedeutet, dass sie auffindbar, zugänglich, interoperabel und wiederverwendbar (aus dem Englischen: findable, accessible, interoperable, reusable, FAIR) sind und öffentlich zugänglich sind. Auf diese Weise können Forscher oder andere Organisationen die Software und die Datensätze frei nach ihren Bedürfnissen nutzen und anpassen. Da der Quellcode der Software öffentlich zugänglich ist, hat sie das Potenzial, von Beiträgen und Vorschlägen der Nutzergemeinschaft zu profitieren. In Zukunft kann die Anwendung DECIMER.ai um die Fähigkeit erweitert werden, textuelle Informationen zu verarbeiten. Darüber hinaus kann die hier vorgestellte Software zur Extraktion von Wissen aus der chemischen Literatur in die Einreichungspipelines offener Datenbanken integriert werden, wo ein menschlicher Kurator lediglich die Informationen validieren muss. Die hier vorgestellte Arbeit ist ein Beitrag zur offenen Verfügbarkeit von chemischen Daten und hat das Potenzial, sich in Zukunft weiterzuentwickeln.

1 Introduction

In the scientific literature, chemical information is commonly published in human-readable data formats like text and images [1]. As this information cannot be directly interpreted by machines, it is inaccessible for automated processing. An example of information about the chemical compound caffeine [2] in text and image formats as they are commonly found in the scientific literature is given in Figure 1.

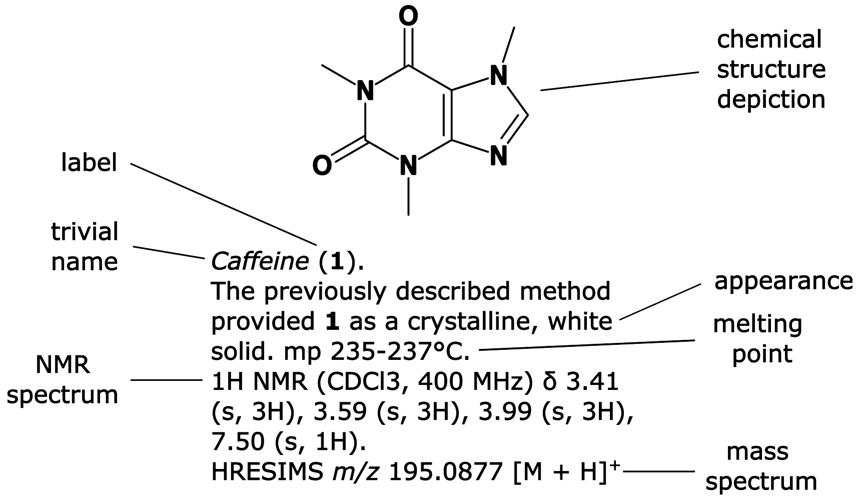

chemical structure depiction
label
trivial name
Caffeine (1).
The previously described method provided 1 as a crystalline, white solid. mp 235-237°C.
1H NMR (CDCl3, 400 MHz) δ 3.41 (s, 3H), 3.59 (s, 3H), 3.99 (s, 3H), 7.50 (s, 1H).
HRESIMS m/z 195.0877 [M + H]+
appearance
melting point
NMR spectrum
mass spectrum

Figure 1: Exemplary information about a chemical compound [2] presented as it is commonly published in the scientific literature

The work presented within this thesis deals with the extraction of chemical information from the scientific literature using deep learning methods. The focus of this thesis is the extraction of molecular structures from images in the literature.

Extracting chemical structures from images in the chemical literature and translating them into machine-readable representations requires multiple steps. The first step is to recognise and segment the structures. Therefore, the first aim of this thesis is to develop a deep learning-based segmentation tool for chemical structure depictions (Publication A). Then, the images need to be processed using an Optical Chemical Structure Recognition (OCSR) application that translates images of chemical structures into machine-readable representations [3]. Accordingly, the second aim of this thesis is the further development of the open-source OCSR application DECIMER (Deep lEarning for Chemical IMagE Recognition) Image Transformer [4, 5] to enable the reliable automated interpretation of chemical structure depictions (Publication E). This thesis also aims to combine the segmentation and OCSR tools in a user interface application that enables the automated extraction of chemical structures from the literature for users without a programming background (Publication E).

The software developed during the work presented herein is based on deep neural network architectures and learns to process information based on the data they are trained on. Diverse training data is necessary to develop a deep learning-based OCSR application that works reliably on the various types of chemical structure depictions in the literature. Therefore, the third aim of this thesis is to develop and implement a diversification strategy for generating chemical structure depictions (Publication B).

All applications herein aim to make previously unavailable chemical information available in structured data formats. To show how this positively affects the research based on data-driven applications, the progress that has been enabled by combining the open availability of chemical data and artificial intelligence in the field of molecular informatics is reviewed (Publication D).

Hand-drawn chemical structure depictions are difficult to interpret for machines due to the various individual drawing styles. In order to enable a systematic performance evaluation of different OCSR methods on hand-drawn chemical structure depictions, this thesis aims at creating a dataset of hand-drawn structure depictions (Publication C).

All applications and datasets developed during the work described herein have already been made public. The source code of the applications is openly available under permissive licenses. The datasets have been published following the FAIR data standards, meaning they are findable, accessible, interoperable and reusable [6].

1.1 Representations of chemical structures

The extraction of chemical information from the literature includes the translation of unstructured representations of chemical structures into machine-readable representations. In cheminformatics, chemical structures are commonly represented as chemical graphs. A graph is an abstract mathematical construct of elements and their connections [7]. The elements are referred to as vertices or nodes, and the connections as edges. In the context of a molecular structure, the atoms are represented by nodes and the bonds are represented by edges [8].

Several commonly used chemical table file formats encode chemical graphs in a tabular format. The core component of these file formats is a connection table which consists of different blocks that contain information about the atoms and bonds in a given molecule [9]. This way, chemical table files contain structured representations of chemical structures.

There is a variety of textual representations of chemical structures that encode the chemical graph. The International Union of Pure and Applied Chemistry (IUPAC) has developed a system for naming organic [10] and inorganic [11] chemical compounds. Images with depictions of chemical structures, IUPAC names and trivial names are representations designed to enable communication between chemists, but they are not meant to be machine-readable. In the case of IUPAC names and structure depictions, they contain the structural information in a complicated, encoded way, making the automated interpretation difficult. In the case of trivial names, the structural information is not contained in the name at all. For example, the name *lysobacteramide A* is used to refer to a natural product that has been extracted from cultures of the bacterium *Lysobacter enzymogenes* C3 [12].

The Simplified Molecular Input Line Entry System (SMILES) is a textual representation that encodes molecular graphs in a machine- and human-readable manner [13]. The International Chemical Identifier (InChI) [14] has been designed as another identifier that can be interpreted by humans and machines that encodes the structural information. The InChIKey is a hashed 27-character version of the InChI [15]. It is meant to be used for indexing chemical databases with unique identifiers, and the molecular graph cannot be restored from an InChIKey [16].

When processing text in an automated manner, tokenisation, the process of splitting the input text into meaningful units (tokens), is a fundamental step [17]. For example, meaningful tokens could be words or syllables when processing text is written in English. When dealing with textual molecular representations, referring to structural units like bonds and atoms as tokens make sense when dealing with textual molecular representations. The sensible tokenisation of SMILES and InChI strings can be difficult, leading to the prediction of invalid structures or wrong syntax when using deep learning models with tokenised textual chemical structure representations as input or output [18]. Consequently, DeepSMILES [19] and SELF-referencIng Embedded Strings (SELFIES) [20] have been developed as molecular string representations that can be split into meaningful tokens so that they only yield valid structures when being used with deep learning models. A study that compared the performance of deep learning models on different tasks when trained with SMILES, DeepSMILES or SELFIES found that the proportion of predicted invalid structures is significantly reduced when using DeepSMILES and it vanishes completely when using SELFIES. Nevertheless, the models trained using SMILES yielded the highest proportion of accurate predictions in this case study [21].

Chemical structures are also commonly depicted as 2-dimensional bitmap images. Although they follow a clear system that communicates the molecular structure to a human reader, a machine cannot easily interpret the grid of pixel values representing the depicted structure. The molecules are often depicted as Markush structures containing R-group variables like 'R_1' or 'X' linked to structural elements defined in separate labels or tables. This way, one depiction can encode multiple molecular structures [22].

The representations of chemical structures in the chemical literature are intended for human interpretation. Structures are mostly referred to by their trivial names or sometimes IUPAC names in the text and are depicted in images. An example of this is given in Figure 2. Here, the breakdown of the prodrug Molnupiravir [23] is presented. The chemical structures are presented as 2D depictions, trivial names (eg. Molnupiravir) or other identifiers (eg. EIDD-1931). EIDD is an abbreviation for the Emory Institute for Drug

Development. Hence, the molecular graph cannot be derived from the names without retrieving further information.

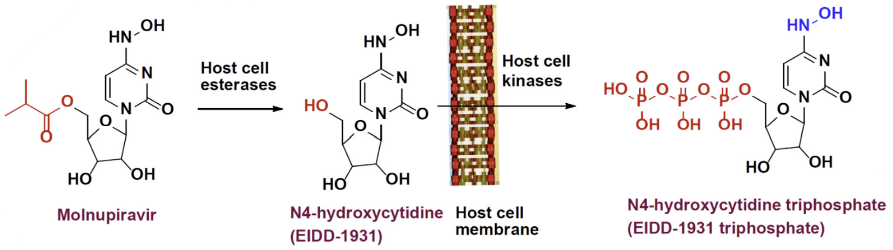


Figure 2: Breakdown of the prodrug molnupiravir as an example of the representation of chemical structures in the literature (image has been taken from [23]).

Molecular fingerprints are another common machine-readable encoding of molecular structures. A molecular fingerprint is a bit-vector where each position indicates the presence or absence of a feature in the encoded molecule [24]. Substructure-key-based fingerprints are binary vectors where each position represents a predefined structural element in the molecule. The structural elements list differs for every type of fingerprint; hence, different fingerprints are helpful for different applications [25]. Examples of substructure-key-based fingerprints are the PubChem fingerprint (length: 881 bits) [26] and the Molecular Access System (MACCS) fingerprint (length: 116 or 960 bits) [27], which is commonly used for drug discovery and virtual screening [25]. In contrast to substructure-key-based fingerprints, path-based or topological fingerprints are generated by determining the encoded fragments following a path up to a given number of bonds from every atom. The fragments are then hashed to allocate a position in the fingerprint. An example of a commonly used path-based fingerprint is the Daylight fingerprint (length: 2048 bits). Circular fingerprints follow a similar principle as path-based fingerprints. Still, instead of following a linear path, the circular environment of each atom is analysed with a radius of a given number of bonds [25]. The extended connectivity fingerprint (ECFP)[28], which is based on the Morgan algorithm [29], is the most widely used type of circular fingerprint [25]. Like substructure-key-based fingerprints, pharmacophoric fingerprints are based on a predefined list of features. Still, these features are not necessarily substructures but can represent 3-dimensional information necessary to encode the activity against a given biological target [30]. There are various ways of representing molecular structures as bit-vectors, and choosing the right one for specific applications can be difficult. To address this problem, Sandfort et al. suggest the concatenation of multiple fingerprints. They have demonstrated that the 71,375-dimensional bit-vector that results from the concatenation of 24 other fingerprints can be successfully used for quantitative structure-property relationship (QSPR) and quantitative structure-activity relationship (QSAR) tasks [31, 32]. An advantage of molecular fingerprints is their fixed size which facilitates processing them using neural networks which require inputs with fixed sizes [33]. Another application is the determination of the similarity between molecules with metrics like the Tanimoto similarity [34], the Dice similarity [35] and the Cosine similarity [36] based on fingerprints. For example, the Tanimoto similarity between two bit-vectors A and B is defined as the ratio of the intersection over the union of the two vectors [34]:

$$T(A,B) = \frac{|A \cap B|}{|A \cup B|} = \frac{N_{1,A,B}}{N_{1,A} + N_{1,B} - N_{1,A,B}} \tag{1}$$

where $N_{1,A,B}$ is the number of bits equal to 1 in both vectors, $N_{1,A}$ is the number of bits equal to 1 in the input vector A, and $N_{1,B}$ is the number of bits equal to 1 in the input vector B. The resulting value lies in a range between 0 and 1 where 0 means that none of the structural elements encoded in the fingerprints occurs in both molecules. In contrast, a value of 1 means that all structural elements occur in both of them.

These metrics can, for example, be used to pick subsets from large sets of molecules that represent the diversity of molecular structures in the original set. This can be done using the MaxMin algorithm, which begins with initialising a subset with a single seed compound from the larger set of molecules and removing this compound from the original set. In the second step, the dissimilarity between each compound in the subset and the remaining compounds in the original set is determined. The smallest dissimilarity values are retained for each compound in the original set. In the third step, the molecule from the original set with the largest retained dissimilarity value is moved to the subset. The second and third steps are repeated until the subset has reached the desired size [37].

1.2 Machine Learning in chemistry

Machine learning systems can learn to solve specific tasks by inferring abstract rules and patterns from given information. That means that these systems do not rely on hard-coded rules to process information, but they learn to process information based on data they are exposed to during a training process [38].

Machine learning methods can be categorised into supervised learning, unsupervised learning and reinforcement learning [39]. A supervised learning system learns to map a set of input variables to a set of output variables to be used to process previously unseen input variables in a desired way [40]. During training, the model's parameters are modified to minimise the resulting value of a loss function that describes the difference between the predicted and the actual output values in the training data [41]. An example of a supervised machine learning model is the multiple linear regression model whose parameters are adapted to minimise the mean squared error loss of the predicted values [42]. In contrast, an unsupervised learning system learns from given data without a set of target variables [43]. For example, k-means clustering is a method that enables dividing a set of data points into a given number of clusters by minimising the sum of the squared distances between the data points within a cluster and the geometric centre of the cluster. During this procedure, the position of the cluster centres and the assignment of data points to the clusters is modified step-wise until the sum of the squared distances converges [44]. Reinforcement learning means that an agent acts in a dynamic environment where it is either penalised or rewarded for its actions. It learns to adapt its actions to maximise the reward function [45]. An example that caught public attention is the system AlphaGo Zero, which achieves superhuman performance in the game Go without having been exposed to any human knowledge during training [46].

Deep learning is a specific area in the field of machine learning that uses deep neural networks to solve specific tasks [47].In 1958, Frank Rosenblatt published the concept of a Perceptron, an information processing system based on the idea of neural information processing in the human brain [48]. The Perceptron, also called single-layer Perceptron, is regarded as the simplest form of a neural network [49]. It is a binary classification system that produces a result y by applying a threshold function with a given threshold ϕ to the sum of the dot product of a vector of input values x and a vector of Perceptron weights w and a constant bias b that is adapted during training [50]. The dot product $x \cdot w$ can be described as the weighted sum of all input values x_i with the weights w_i. This is described in Equation 2 and illustrated in Figure 3.

$$y = \begin{cases} 1 & \text{if } \sum_{i=1}^{N} w_i x_i + b \geq \phi \\ 0 & \text{if } \sum_{i=1}^{N} w_i x_i + b \leq \phi \end{cases} \tag{2}$$

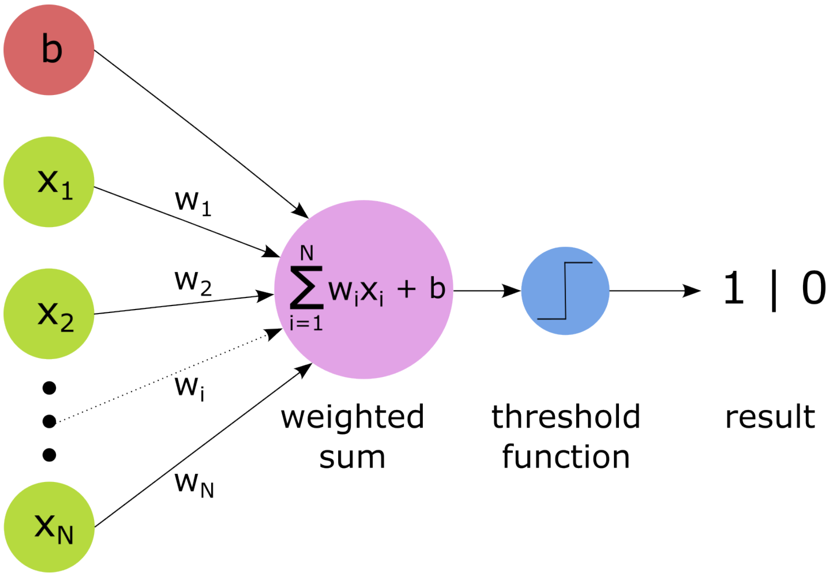


Figure 3: Illustration of a binary classification with a single-layer Perceptron. A threshold function is applied to the sum of a bias b and the dot product of a vector x containing all input values and a vector w with the trained weights of the model. Based on a threshold ϕ, the value 1 or 0 is returned.

A Perceptron unit can be described as a neuron in a neural network architecture. Here, the activation function applied to the dot product of the input vector x and the weight vector w then determines the activation of this neuron. In the example above, the activation function is a threshold function, but various activation functions can be applied [51]. For instance, in contrast to the threshold function, which returns discrete values, a sigmoid function $f_{sig}(x)$ maps the given input value to a continuous value range between 0 and 1 [52]:

$$f_{sig} = \frac{1}{1 + e^{-x}} \tag{3}$$

Based on this information, the output y of a neuron with any activation function f_a in a neural network can be described as follows:

$$y = f_a\left(\sum_{i=1}^{N} w_i x_i + b\right) \tag{4}$$

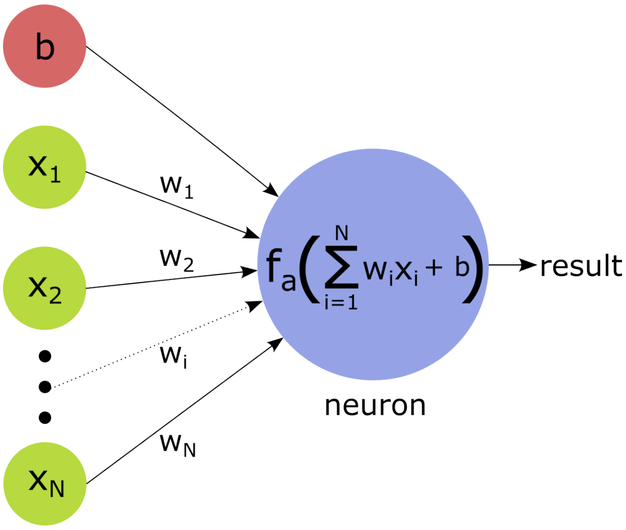


Figure 4: Schematic illustration of how a neuron in a neural network produces a result by applying an activation function f_a to the sum of a bias b and the dot product of a vector x containing all input values and a vector w with the trained weights of the neuron (see Equation 4).

When Perceptron units are arranged in multiple layers, they form a multilayer Perceptron (see Figure 5). In this fully connected neural network architecture, the input layer processes an input vector x, passing each value on to every neuron in the first hidden layer. Here, each neuron processes the given information as described in Equation 4. Each value from the resulting output vector h is fed to each neuron in the next hidden layer. This procedure continues until the information has been fed through the complete network, yielding a vector y of output values [52].

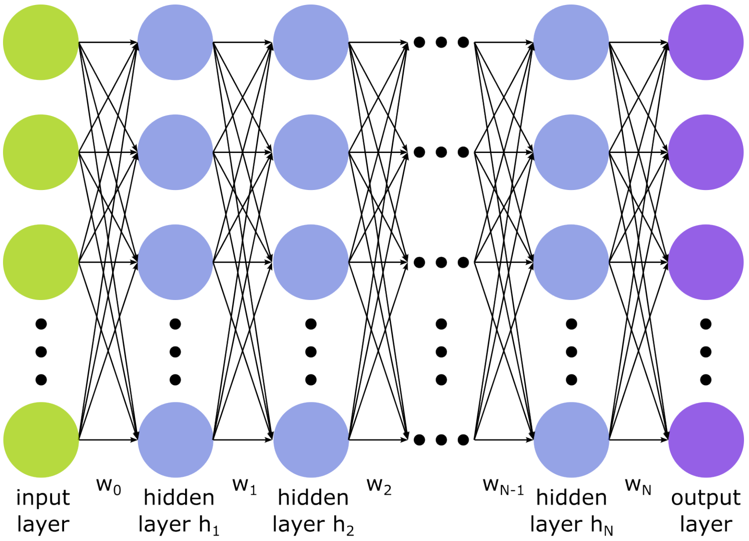


Figure 5: Schematic illustration of a multilayer Perceptron

In a feed-forward network like the multilayer Perceptron, information is passed forward from the input layer to the output layer. In a recurrent neural network (RNN), information is additionally passed from hidden layers to previous ones, to themselves, or to separate context units that pass it on to hidden layers. This enables RNN to process sequential data like time series or language [53]. An RNN is capable of mapping an input sequence x to an output sequence y by processing them using a set of hidden layers. Every output value y_t at a given time t depends on the hidden state s_t which depends on the corresponding input value x_t and the previous hidden state which again depends on the previous hidden state s_{t-1} and the previous input x_{t-1}. This way, every output y_t indirectly depends on the current input x_t and every previous input. The weight matrices that are used to compute the hidden states from the inputs (U), the output from the hidden state (V), and the next hidden state based on the previous one (W) are the same at every time step [54]. The information processing in an RNN is illustrated in Figure 6.

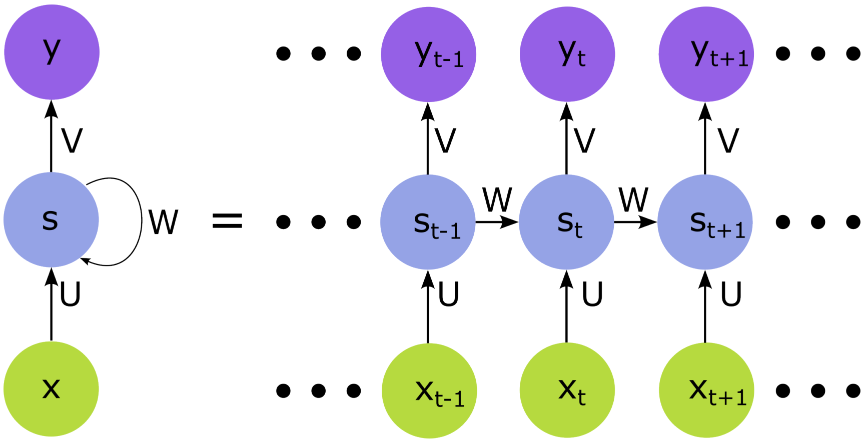


Figure 6: Representation of the structure of a recurrent neural network (left) and the same representation unfolded in time (right). The input sequence x is mapped to the output sequence y. Every output y_t at the time t depends on the corresponding hidden state s_t, which depends on the current input x_t and the previous hidden state s_{t-1}. The weight matrices U, W and V are the same at every time step [54].

In a bidirectional RNN, the hidden states s are split into two independent components: the forward states, which are influenced by the current input and the previous hidden state and the backward states, which are influenced by future hidden states (Figure 7). This way, every output value is influenced by previous and future hidden states, leading to increased performance compared to conventional RNN [55].

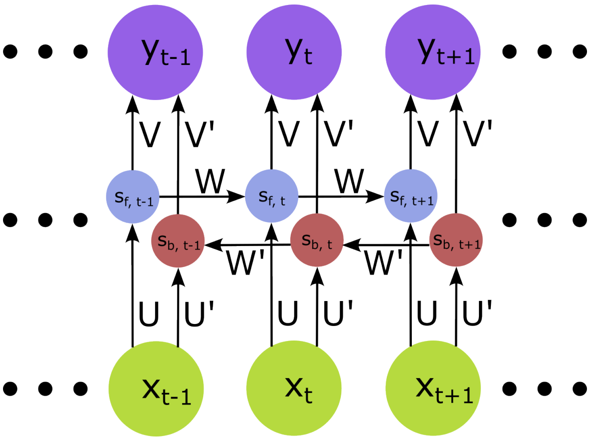


Figure 7: Representation of a bidirectional recurrent neural network unfolded in time. The input sequence x is mapped to the output sequence y. Every output y_t at the time t depends on the forward hidden state $s_{f,t}$ and the backward hidden state $s_{b,t}$. $s_{f,t}$ depends on the current input x_t and the previous forward hidden state $s_{f,t-1}$ and $s_{b,t}$ depends on the current input x_t and the next backward hidden state $s_{b,t+1}$. The weight matrices U, U', W, W', V and V' are the same at every time step [54].

Unfortunately, while the immediate context of an element in a sequence can be modelled well using RNNs, in practice, RNN are not capable of learning long-term dependencies [56]. To address this problem, Hochreiter and Schmidhuber have introduced the concept of Long Short-Term Memory (LSTM) networks [57]. Each cell in an LSTM network has an input gate that decides which information is saved in the cell state and an output gate that determines which information output is passed on based on the cell state [58]. A forget gate was later added to the LSTM cell to improve the selection of information passed on through the network. The forget gate decides what information is removed from the current cell state [59]. Gated Recurrent Unit (GRU) cells have been introduced to reduce the increased computational complexity of information processing in an LSTM cell compared to a conventional RNN cell [60]. A GRU cell has two gates: an update gate to decide what information is passed on and a forget gate to determine what information is removed [58].

Typically, an RNN-based machine translation system follows an encoder-decoder architecture. The input sequence is processed using a bidirectional RNN, and the last output is fed as a context vector to every node in a bidirectional RNN decoder which produces the output sequence. This leads to a performance loss with an increasing input sequence length as more information needs to be compressed in a fixed-length context vector. To address this problem, Bahdanau et al. introduced the concept of attention. Instead of using the output of the last encoder RNN cell, the context vector defined is the weighted sum of the attention weights and the output values y of the encoder at every point in time. The attention weights are parameters that are adjusted during training. This way, the model has access to all relevant aspects of the input sequence when generating each part of the output sequence and can focus on certain parts more than others based on the current state [61].

In the field of chemical information extraction, recurrent neural networks have been used in chemical named entity recognition (NER). For example, the application ChemListem uses a bidirectional LSTM network to recognise chemical names in text sequences [62]. Another application is the design of new drugs. Segler et al. were able to demonstrate that an LSTM network can be used to generate SMILES representations of molecules with desired properties and biological activities that were not included in the training data [63]. Rajan et al. have used an encoder-decoder network with GRU cells and an attention mechanism to translate IUPAC names to SMILES representations and vice versa with their application SMILES to IUPAC Translator (STOUT) [64]. Winter et al. conducted similar research when publishing an RNN-based encoder-decoder architecture for the translation of IUPAC names to SMILES representations of chemical structures. The latent feature vector that is produced by the encoder can be utilised as a meaningful chemical representation for QSPR tasks and is referred to as a continuous and data-driven molecular descriptor (CDDD)[65].

In 2017, the Google Brain team introduced a novel architecture for processing sequences

- the transformer [66]. The transformer model is an encoder-decoder based on attention. The encoder and the decoder block consist of multiple stacked encoders or decoders. Each encoder block consists of a self-attention layer and a feed-forward network. Each decoder block consists of a self-attention layer, an encoder-decoder attention layer and a feed-forward network. As there is no recurrence in the transformer architecture, a positional encoding is added to the input embeddings at the beginning of the encoder and the decoder stacks. Each attention and feed-forward layer in the encoder and the decoder block has a residual connection around it which is used with a given dropout probability during training. The model architecture is depicted in Figure 8.

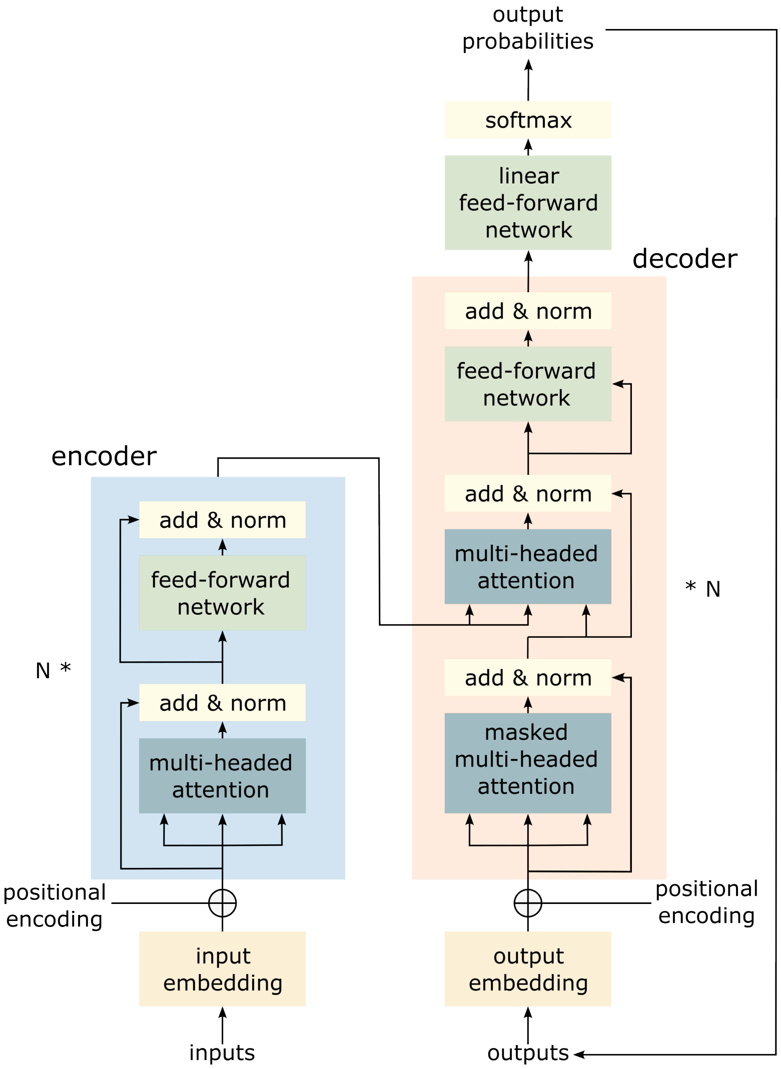


Figure 8: Transformer model architecture according to Vaswani et al. [66]

The self-attention layers in the stacked encoder blocks form a mechanism that defines the influence of all tokens in the input sentence on the encoding of a specific token. For example, while encoding the SMILES string 'CC(O)C' with character-based tokenisation in the first encoder block, it would make sense to consider the opening and closing brackets while encoding the ’O’ as only all three tokens together represent a hydroxy group. The attention that is computed here is referred to as scaled dot product attention. For each element in the input of a self-attention layer, a query vector q, a key vector k and a value vector v are generated by multiplying the input with one of three matrices (W^Q, W^K, W^V) whose values are determined during training. Then, for each element in the input, a score for every other element in the input is computed. Therefore, the dot products of the query vector of the element being encoded and the key vector of every other element are computed and divided by the square root of the dimension of the key vector $\sqrt{d_k}$. Using the softmax function, the resulting scores are normalised to a value range between 0 and 1. To compute the scaled dot-product attention, each normalised score is multiplied with the corresponding value vector and the values of the resulting vector are summed up. This way, while processing one element in the input of the self-attention layer, an attention value for itself and every other element is generated.

In the practical implementation, these operations are not executed one by one. Still, the vectors for keys, queries and values are summarised in the matrices Q, K and V, and the self-attention is determined as described in Equation 5 [66]:

$$scaled\_dot\_product\_attention(Q,K,V) = softmax\left(\frac{QK^T}{\sqrt{d_k}}\right)V \tag{5}$$

At every step, the scaled dot-product attention is computed multiple times in parallel, which is why it is referred to as a multi-headed attention mechanism. Each attention head can learn to attend to different relationships in the input sequence. For example, in the SMILES representation of isopropanol above, while encoding the ’O’, one attention head might attend to the brackets that define the branched character of the hydroxy group (’CC(O)C’) while another one might attend to the ’C’ that represents the connected carbon atom (’CC(O)C’). The resulting attention matrices are concatenated and multiplied with a different matrix W^O whose values are determined during training. The encoder-decoder attention in the decoders works like the above-described multi-headed scaled dot-product attention. The only difference is that it takes key and value matrices K and V from the last encoder and the query matrix from the previous layer in the decoder stack. A linear feed-forward network and a softmax layer then process the output of the last decoder. This produces a vector of the dimension of the output vocabulary and contains values between 0 and 1, representing prediction probabilities for each possible output token. The token with the highest predicted probability is returned. At every time step, a token is predicted this way. The already predicted tokens are then the input for the decoder stack at the next step. Here, it is important that the decoder can only attend to predicted outputs at previous time steps. Therefore, future positions are masked in all attention layers in the decoders. [66]

Since the concept of transformers has been published, they have replaced the previous

RNN-based models in the field of natural language processing (NLP) [67]. Models based on the concept of Bidirectional Encoder Representations from Transformers (BERT) [68] achieve state-of-the-art results in a variety of NLP tasks [69]. Large language models (LLMs) like OpenAI's GPT-3 (Generative Pretrained Transformer 3) [70] or Meta's LLaMA (Large Language Model Meta AI) [71] can be used for a variety of text generation applications [72].

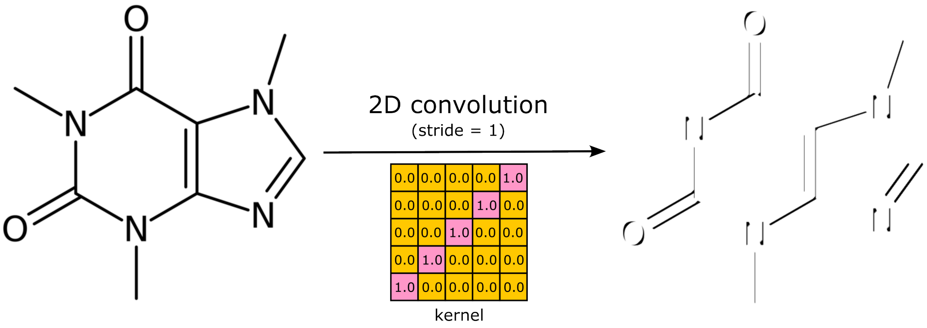
In chemical literature mining, transformers have been used to extract knowledge from text inputs. For example, a BERT-based model is used for the chemical NER in the literature mining application ChemDataExtractor 2.1 and can recognise organic and inorganic chemical names [73]. The applications BatteryDataExtractor and MatSciBERT use BERT models to extract information from different domains of material science [74, 75]. Some deep learning-based applications that translate images of chemical structures into machine-readable representations use a transformer model as their decoder (see Section 1.3).

In this work, a transformer model is used as a decoder in the application DECIMER Image Transformer (see Publication E). Additionally, Publication D describes the usage of transformers for the prediction of reaction outcomes [76], yields [77], retrosynthesis planning [78] and other use-cases in the field of synthetic chemistry.

In the field of computer vision and image processing, a different class of neural networks is commonly used - convolutional neural networks (CNN). A CNN is a partially connected feed-forward network that typically consists of convolutional layers, pooling layers and fully connected layers [79]. The first convolutional neural network architecture, the Neocognitron, was introduced by Fukushima in 1988 [80].

In theory, images can be processed using a multilayer Perceptron, but the resulting high number of weights leads to a computationally expensive, inefficient system. For example, processing information from a grayscale input image with a resolution of 64 x 64 pixels to a single hidden layer of the exact dimensions as the input image would result in $(64 * 64)^2 = 16{,}777{,}216$ weights. If every neuron is only connected to 5 x 5 pixels from the previous layer, the amount of weights in the example above is decreased to $(5 * 5) * 64 * 64 = 102{,}400$. Following the assumption that image features are relevant independent of their position in the image, these weights can be kept constant for all connections. In the example, the number of unique weights that need to be trained is then reduced to $5 * 5 = 25$. A convolutional layer is a partially connected layer that implements this type of connection. The effect resembles a filter window or kernel that slides over the image and defines the input of the neurons in the next layer based on fixed weights. The kernel shape, the number of kernels, and the stride (the kernel's step size to slide over the input image) are important hyperparameters that define a convolutional layer [81]. For example, processing an image with the shape 128 x 128 x 3 is processed with a convolutional layer with a kernel shape of 4 x 4 x 3, a stride of 1 and 10 convolutional kernels would result in a feature map with the shape 128 x 128 x 10. In this case, padding must be added to the original image to process pixels at the edge of the input image. Splitting the feature map along the third axis results in 10 images with the shape of the original image, where certain features are included differently based on the kernel weights.

The application of a convolutional layer to a grayscale image (512 x 512 pixels) with a 5 x 5 kernel and a stride of 1 is illustrated in Figure 9. In this example, the kernel weights are defined as 1 along a diagonal line, and all other weights are defined as 0. This way, elements with the same diagonal orientation in the original image are represented more than others in the convolved image.



0.0	0.0	0.0	0.0	1.0
0.0	0.0	0.0	1.0	0.0
0.0	0.0	1.0	0.0	0.0
0.0	1.0	0.0	0.0	0.0
1.0	0.0	0.0	0.0	0.0

Figure 9: Illustration of a 2-dimensional convolution of an image that contains a depiction of the structure of caffeine.

Pooling layers are downsampling layers that are typically placed after the convolutional layers. Their filter size and stride characterise them. The filter size defines the size of a filter that is moved over a given feature matrix. The stride is the step size that is used while moving the filter. Then, depending on the type of pooling, only one signal is returned for every position of the filter in the input. For example, max pooling returns only the highest value for each filter position. The purpose of pooling is the reduction of the complexity of information without losing essential features before passing it on to the next layer [82]. An exemplary representation of max and average pooling is presented in Figure 10.

Typically, CNNs are built using multiple blocks of convolutional layers followed by a pooling layer. This way, the network can learn to recognise low-level image features like lines or corners and combine them into representations of high-level features like bonds, atoms or characters in a chemical structure depiction. Deep learning-based image captioning systems combine CNN encoders with RNN or transformer decoders to generate text based on images. Here, the feature map generated based on the image is processed by a sequence model to generate text [83].

The OCSR application Img2Mol uses three blocks of two or three convolutional layers followed by max pooling and a fully connected feed-forward network to generate a feature map based on chemical structure depictions. This feature map is then translated to a

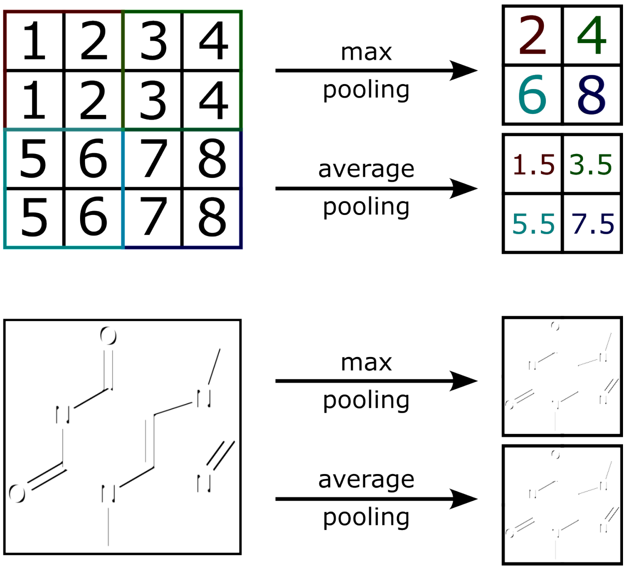


Figure 10: Illustration of max pooling and average pooling with a kernel size and stride of 2 applied to an exemplary matrix (above) and the convolved image from Figure 9 (below).

SMILES string using an RNN decoder [65, 84] (see Section 1.3).

In chemistry, CNNs have been widely used in the field of chemical literature mining. These developments are described in Section 1.3. Some other applications are the detection of ligand-binding sites in visualised 3-dimensional protein structures [85]. A token in a sequence can be represented as a binary vector with a single high bit, a so-called one-hot vector. The vector typically has the size of the token vocabulary, and the 1 encodes which token is represented by the vector [86]. One-hot-encoded SMILES strings have been used as a binary matrix input for CNNs to determine glass transition temperatures of polymers [87], the identification of functional substructures [88] and the prediction health effects of aerosols [88]. This work uses the convolutional neural network architecture EfficientNetV2 [89] as the encoder of the updated DECIMER Image Transformer model presented in Publication E.

Another application of CNNs is instance segmentation. Here, every pixel of every instance of an object in a given image is assigned a separate categorical label [90]. During the work on this thesis, the application DECIMER Segmentation which can segment chemical structure depictions in the scientific literature, has been developed (see Publication A). DECIMER Segmentation uses the Mask Region-based Convolutional Neural Network (Mask R-CNN) architecture. The Mask R-CNN semantic segmentation workflow begins with generating a feature map based on a given image using a CNN. Then, an additional CNN, the region proposal network, generates the proposed region in the feature map that contains information about the objects to be segmented. Based on these region proposals and the feature map, Mask R-CNN generates a bounding box, a mask and a class label for each instance of a segmented object. Here, a mask is a binary matrix with the same shape as the original image where the value 1 labels a pixel as part of the object, while 0 labels the corresponding pixel as not part of the object [91].

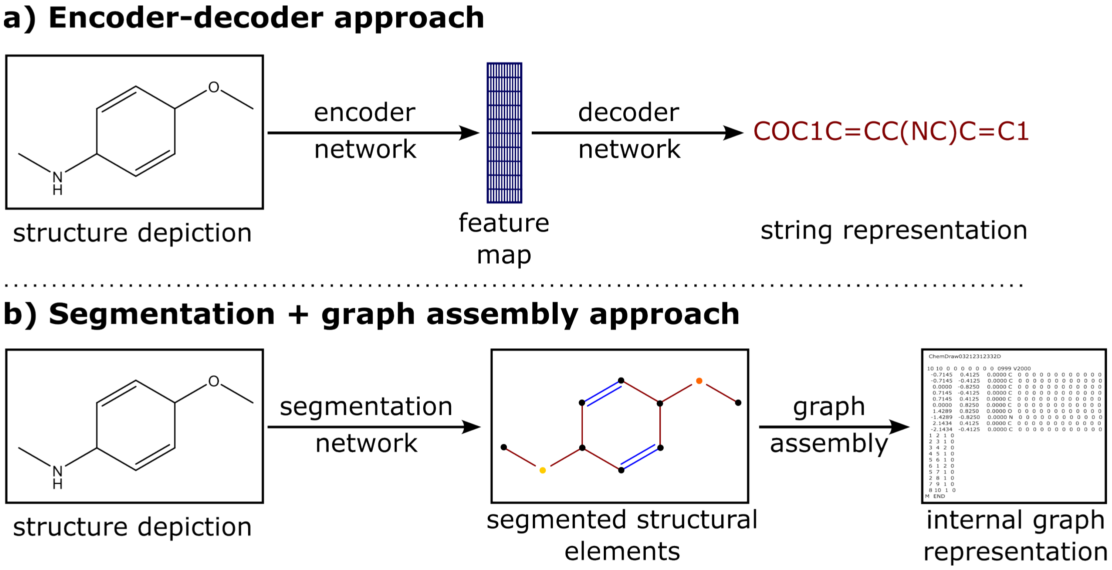
1.3 Optical chemical structure recognition

The translation of depictions of chemical structures into machine-readable representations is commonly referred to as Optical Chemical Structure recognition (OCSR) [3, 92]. The first OCSR software application Kékulé was published in 1992 [93]. Until the first deep learning-based OCSR tool was published in 2019 [94], most OCSR tools approached the task by applying a hard-coded algorithm to the given images [3]. These rule-based tools follow a pre-defined workflow that assembles the molecular graph based on detected lines or vectors and text elements in the binarised image.

To better illustrate this process, this paragraph describes the workflow of the first open-source OCSR tool OSRA that was published in 2009 [95]. First, the input image is converted to grayscale and then binarised. Subsequently, the region in the image that contains the chemical structure is determined based on the proportion of black pixels and the dimensions and the height-to-width ratio of the resulting region. Then, an anisotropic smoothing algorithm is applied to reduce noise. The thickness of each line in the image is reduced to one pixel using a thinning algorithm. The image is then vectorised. Specific control points are interpreted as atoms based on a set of rules. The vectors that connect these control points are accordingly interpreted as bonds. Text elements representing atom or superatom labels or charges are resolved using OCR. The corresponding atoms are labelled as aromatic if a circle is detected inside a ring structure. Similarly, if multiple lines are arranged in parallel within a certain distance, they are interpreted as double or triple bonds and the bond order is modified accordingly. The 75th percentile of all bonds is interpreted as the average bond length in the molecule. When three or more elements are detected within the average bond length and a straight line can be drawn through their centres, they are interpreted as a dashed bond. If the thickness of a line of the average bond length increases linearly, it is interpreted as a wedge bond. A set of rules is in place to identify intersecting lines where the intersection does not represent an atom. Finally, the molecular graph is assembled in a connection table based on the gathered information. Here, known superatom labels are inserted based on the previously recognised text labels. Every input image is processed three times in different resolutions, and the best result is picked based on a confidence estimate generated using an empirically determined function [95].

Apart from OSRA, two more rule-based open-source systems (Imago [96] and Molvec [97]) have been published. Additionally, there is a row of tools and methods that have been described in publications but are only available as commercial tools, like the Chemical Literature Data Extraction (CLiDE) project [98, 99] and ChemOCR [100] or completely unavailable [101–103].

From 2019 on [94], several deep learning-based OCSR methods have been published. Most generate a compressed feature map from a given input image using an encoder network and then generate a molecular string representation from that feature map with a decoder network (see Figure 11 a). Some other approaches are based on the segmentation of different structural elements in a given chemical structure depiction and the subsequent assembly of a molecular graph based on the segmented elements (see Figure 11 b). An overview of deep learning-based OCSR methods is presented in Table 1.



```
ChemDraw03212312332D

10 10  0  0  0  0  0  0  0  0999 V2000
  -0.7145    0.4125    0.0000 C   0  0  0  0  0  0  0  0  0  0  0  0
  -0.7145   -0.4125    0.0000 C   0  0  0  0  0  0  0  0  0  0  0  0
   0.0000   -0.8250    0.0000 C   0  0  0  0  0  0  0  0  0  0  0  0
   0.7145   -0.4125    0.0000 C   0  0  0  0  0  0  0  0  0  0  0  0
   0.7145    0.4125    0.0000 C   0  0  0  0  0  0  0  0  0  0  0  0
   0.0000    0.8250    0.0000 C   0  0  0  0  0  0  0  0  0  0  0  0
   1.4289    0.8250    0.0000 O   0  0  0  0  0  0  0  0  0  0  0  0
  -1.4289   -0.8250    0.0000 N   0  0  0  0  0  0  0  0  0  0  0  0
   2.1434    0.4125    0.0000 C   0  0  0  0  0  0  0  0  0  0  0  0
  -2.1434   -0.4125    0.0000 C   0  0  0  0  0  0  0  0  0  0  0  0
  1  2  1  0
  2  3  1  0
  3  4  2  0
  4  5  1  0
  5  6  1  0
  6  1  2  0
  5  7  1  0
  2  8  1  0
  7  9  1  0
  8 10  1  0
M  END
```

Figure 11: Schematic overview of the two approaches used by deep learning-based OCSR applications. a) Generation of a feature map with an encoder which is then decoded into a string representation of the depicted molecule. b) Segmentation of structural elements in the structure depiction and subsequent assembly of the molecular graph.

Staker et al. were the first to propose an encoder-decoder architecture where an image with a chemical structure depiction is processed with a CNN architecture to generate a state vector which is then processed by an LSTM network with an attention mechanism to generate SMILES strings [94]. The method is referred to as molecular structure extraction from documents using deep learning (MSE-DUDL). The model has been trained on images from the United States Patent and Trademark Office (USPTO) and artificial depictions that have been generated using Indigo [104]. The Publication also presents the segmentation of chemical structure depictions based on the U-Net architecture [105]. The source code and the trained model weights are not publicly available.

Table 1: Overview of deep learning-based OCSR applications

Application	**Approach**	**Reconstructed molecular representation**	**Availability, license**
MSE-DUDL	Encoder-decoder architecture	SMILES	Closed-source, unavailable
ChemGrapher	Segmentation of structural elements and graph assembly	Internal graph representation	Open-source, no license
Image2SMILES	Encoder-decoder architecture	SMILES	Closed-source, unavailable
DECIMER Image Transformer (1.0)	Encoder-decoder architecture	SELFIES	Open-source, MIT
Img2Mol	Encoder-decoder architecture	SMILES	Open-source, CC BY-NC 4.0
MICER	Encoder-decoder architecture	SMILES	Open-source, MIT, model weights unavailable
ABC-Net	Segmentation of structural elements and graph assembly	Internal graph representation	Closed-source, unavailable
ChemPix	Encoder-decoder architecture	SMILES	Open-source, Apache 2.0
SwinOCSR	Encoder-decoder architecture	DeepSMILES	Open-source, no license
MolMiner	Segmentation of structural elements and graph assembly	Internal graph representation	Closed-source, restricted access for registered users
MolScribe	Segmentation of structural elements and graph assembly	Internal graph representation	Open-source, MIT

ChemGrapher uses a different approach by splitting the OCSR workflow into two tasks: semantic segmentation and classification. First, the semantic segmentation step determines the position of all atoms, bonds and charges in the input image. This is done using a Dense Prediction CNN based on dilated convolution [106]. In the second step, all segmented elements are labelled with a specific atom, bond and charge type using three classification networks. This information is then used to assemble the chemical graph. The developers of ChemGrapher use the cheminformatics library RDKit to generate the training data, which requires images where every pixel is labelled [107]. The source code of the application and the trained model weights are publicly available [108].

Image2SMILES uses an encoder based on the CNN architecture ResNet-50 [109] and a transformer decoder to generate strings with a modified SMILES syntax (FG-SMILES) based on chemical structure depictions [110]. At the time of publication, the capability to read R-group labels was an outstanding factor. The training data generator, which is based on the depiction functionalities of RDKit, is openly available [111]. Still, the source code of the application itself and the trained model weights have not been published.

DECIMER Image Transformer (1.0) uses a pre-trained EfficientNetV1 B-3 [112] encoder and a Transformer decoder for the translation of chemical structure depictions into SMILES strings. The open-source tool has been trained on pairs of SELFIES and structure depictions generated using the cheminformatics library Chemistry Development Kit (CDK) [113] with the application of mild image augmentations. This version of DECIMER Image Transformer yielded 89 % of perfect predictions on the in-domain test data that includes chiral molecules and ions but is not capable of interpreting R-group variables in Markush structures [5]. All datasets, the source code, and the DECIMER Image Transformer model weights are publicly available [114].

Clévert et al. trained a CNN encoder to generate the previously mentioned CDDD based on images of chemical structures [84]. In combination with the decoder from the previous work [65], the resulting system *Img2Mol* is capable of translating images of chemical structures to SMILES strings. The model has been trained on images with chemical structure depictions that have been generated using RDKit [115], Indigo [104] and the proprietary application OEChem from OpenEye [116]. As the CDDD cannot encode chirality or Markush structures [65], Img2Mol cannot translate images containing these features. The source code and the trained model weights are openly available for non-commercial usage [117], but the training data generation pipeline is unavailable.

The Molecular Image Captioner (MICER) is based on a CNN encoder and an LSTM network with an attention mechanism as a decoder for the generation of SMILES strings from structure depictions. The structure depictions used to train the model have been generated using RDKit [115] and Indigo [104]. The source code of the implementation of the model is openly available [118], but the trained model weights have not been published.

The Atom and Bond Center Network (ABC-Net) is a fully convolutional neural network with skip connections. Based on a chemical structure depiction, it generates multiple heatmaps that describe the position and type of the atoms and bonds in the image as well as associated properties. The training data has been generated using Indigo [104] and RDKit [115]. The molecular graph is then assembled based on the elements that are represented in the heatmaps [119]. The source code and the model weights of ABC-Net are not publicly available.

ChemPix is an OCSR model that has been specifically developed for the translation of hand-drawn chemical structure depictions. The model architecture is based on a CNN encoder and an LSTM decoder. It has been trained on pairs of images of chemical structure depictions and the corresponding SMILES representations of the depicted molecules. The developers of ChemPix have created an image augmentation pipeline for the training data so that the structure depictions generated with RDKit [115] appear hand-drawn-like, although they have been generated artificially. The usage of ChemPix is limited to molecules that only contain carbon and hydrogen atoms [120]. The source code and the trained model weights are openly available under a permissive license [121].

Xu et al. suggested an encoder-decoder architecture based on a Swin Transformer [122] and a Transformer decoder to generate DeepSMILES based on images of chemical structures. The resulting tool SwinOCSR has been trained on images of chemical structures generated using the CDK and performs well on this specific type of depiction. Still, according to the authors, it tends to fail on images of chemical structures published in the literature [123]. The source code and the trained weights of the model are publicly available [124].

MolMiner combines multiple deep learning-based segmentation models with OCR to detect and recognise bonds and labels in a depicted molecule. These detected elements and their relative positions are then used to reconstruct the molecular graph. Compared to most other OCSR tools, MolMiner can process whole document pages and segment the chemical structure depictions before translating them into a machine-readable representation [125]. The training data has been generated using RDKit [115]. The user interface application is available under a restrictive license and cannot be used to process large data batches. The source code and the models are not openly available.

Like SwinOCSR, MolScribe uses an encoder-decoder architecture based on a Swin Transformer encoder and a Transformer decoder, but it predicts atom labels and their coordinates. A second feed-forward network predicts the bond between every pair of atoms. This way, MolScribe can recover the chemical graph based on depicted molecules. The training data is a mixture of data collected from patent data published by the USPTO and artificially generated depictions created using Indigo [104]. The generation of the coordinates based on the image enables the re-depiction of the resolved molecule in the same way it is depicted in the original image, which facilitates the assessment and correction of the result by a human curator [126]. The source code and the trained models are publicly available under a permissive license [127].

1.4 Scope of this thesis

Typically, chemical information is published in the form of unstructured, human-readable data formats such as text and images. This thesis aims to contribute to the extraction of chemical information from the scientific literature and the recovery of this information in structured data formats using deep learning. The main focus is extracting information from image formats, i.e. the segmentation and recognition of chemical structure depictions in the literature.

Machine learning-based, artificially intelligent applications are becoming more and more important in the field of chemistry. Data-driven applications can only be implemented if data is available that they can learn from. To elucidate the importance of openly available chemical data, the recent advances in molecular informatics made possible by data-driven applications have been reviewed as a part of this thesis (see Publication D).

The tool DECIMER Segmentation has been developed to segment chemical structure depictions from the printed literature. Currently, it is the only deep learning-based open-source application that fulfils this purpose (see Publication A). Additionally, the OCSR application DECIMER Image Transformer has been further developed, has reached state-of-the-art results, and has been published along with the DECIMER Image Classifier, an application for the classification of images that contain chemical structure depictions. All three components have been integrated into the DECIMER.ai web application, the first open-source platform for extracting chemical structures from the printed literature (see Publication E).

The high performance that DECIMER Image Transformer achieves is due to the diversification of the training data implemented in the form of the chemical structure generation tool RanDepict that has been developed as part of this thesis. RanDepict can generate sets of chemical structure depictions with diverse depiction features and has mechanisms implemented to ensure the diversity of the generated datasets (see Publication B).

The DECIMER hand-drawn molecule image dataset is the first openly available diverse set of hand-drawn chemical structure depictions for evaluating the performance of OCSR applications. The dataset comprises 5088 hand-drawn chemical structures and has been published following the FAIR data standards.

All software applications and datasets that have been developed during the work on this thesis are openly available under permissive licenses. As a result, other researchers or organisations are able to use and adapt them according to their needs. The software is designed to be used without further modifications but can be extended if necessary. All publications in this cumulative thesis have been published as open-access articles.

1.4.1 Achievements of data-driven applications in chemistry

The main focus of this thesis is extracting chemical information from the scientific literature to make it available in structured, machine-readable data formats. The motivation for making chemical information publicly accessible in structured formats is described in Publication D, where the progress driven by machine learning-based applications in the field of molecular informatics in recent years is described. The Publication describes the success of AlphaFold with the prediction of 3-dimensional protein structures, breakthroughs in the development of synthesis prediction and retrosynthesis planning systems based on Transformer models and AI-based progress in the field of natural product-based drug discovery. These achievements have only been possible due to the application of data-driven deep learning methods in areas where data has been openly accessible. The progress in chemical literature mining is reviewed as a technology that may enable more data-driven progress in the future. Additionally, openly available toolkits, databases and repositories, and open research management initiatives that support the FAIR data standards are presented.

1.4.2 Development of an application for the segmentation of chemical structure depictions

A fundamental step for extracting information about chemical structures from the chemical literature is the recognition and segmentation of chemical structure depictions on whole pages from scanned documents or articles. Publication A describes the development and performance evaluation of DECIMER Segmentation, a deep learning-based application for segmenting chemical structure depictions from the scientific literature. At its core, DECIMER Segmentation uses a Mask R-CNN segmentation model to generate binary matrices that describe the position of the chemical structures on the processed page. A mask expansion algorithm that traces all pixels that belong to the depictions has been implemented to refine the generated masks. The Mask R-CNN model has been trained on 9992 manually annotated chemical structures on 1820 pages of articles from the Journal of Natural Products but has been shown to perform well on other publisher formats as well (see Publication A). As DECIMER Segmentation is the only openly available deep learning-based solution to this problem, it represents a considerable contribution to the field of chemical literature mining.

1.4.3 Improvement of the OCSR application DECIMER Image Transformer

To improve the performance of the OCSR application DECIMER Image Transformer [5], the model architecture and the training data have been improved (see Publication B and E). The previous model has been replaced with a fully trainable encoder-decoder model based on an EfficientNetV2-M [89] encoder and a Transformer [66] decoder. Additionally, the chemical structure generation application RanDepict has been developed (see Publication B) to generate large sets of diverse chemical structure depictions to train the model with. Previous versions had been trained on depictions rendered with the default settings of the depiction functionalities of the CDK. RanDepict makes use of all available adjustable depiction parameters of the cheminformatics toolkits CDK [113], RDKit [115], Indigo [104] and the Python-based Informatics Kit for Analysing Chemical Units (PIKAChU) [128] to generate diverse training data. Over time, more and more features have been added to the training data to improve the results of DECIMER Image Transformer models that are trained on it. For example, generating sets of Markush structures based on given sets of molecules and their diverse depictions has led to DECIMER Image Transformer being capable of interpreting depictions with different types of R group variables. The latest version of DECIMER Image Transformer has been trained on more than 450 million pairs of chemical structure depictions and the corresponding SMILES representations. It yields very competitive results in the comparative performance analysis presented in Publication E. It is a reliable and robust open-source application for translating chemical structure depictions into machine-readable representations.

1.4.4 Creation of a benchmark dataset for hand-drawn chemical structure depictions

During the development of DECIMER Image Transformer, it became apparent that the models trained on depictions generated with RanDepict are partially capable of interpreting hand-drawn chemical structure depictions. However, no hand-drawn structures are included in the training data. As no diverse set of hand-drawn chemical structure images was available to evaluate the performance systematically, the DECIMER hand-drawn molecule image dataset was created in collaboration with 24 volunteers from the Westphalian University of Applied Sciences Recklinghausen, Germany. The dataset consists of 5088 hand-drawn chemical structures. It follows the FAIR data standards and is openly available as a contribution to the field of hand-drawn OCSR.

1.4.5 Creation of a platform for the extraction of chemical structure information from the literature

After developing robust solutions for the segmentation and the interpretation of chemical structure depictions in the printed literature, the next step was their integration into a chemical literature mining system with a graphical user interface. This has been implemented by developing the DECIMER.ai web application described in Publication E. This platform makes the previously developed applications DECIMER Segmentation and DECIMER Image Transformer accessible to a larger group of end-users because operating it does not require programming skills. Users can load a Portable Document Format (PDF) document; all chemical structures are automatically segmented and interpreted. The resolved SMILES representations of the depicted molecules are shown and loaded into an integrated molecular structure editor. Here, they can be edited before downloading the segmented images and the chemical table files. This enables a human curator to make adjustments if necessary. A user can alternatively directly upload chemical structure depictions to generate machine-readable representations. The DECIMER Image Classifier, capable of distinguishing between chemical structure depictions and non-chemical images, has been developed and integrated into DECIMER.ai to produce a warning when users upload non-chemical images. As an open state-of-the-art platform for extracting chemical structures from the scientific literature, DECIMER.ai contributes significantly to the field of chemical literature mining.

2 Publications

2.1 Publication A: DECIMER-Segmentation: Automated extraction of chemical structure depictions from scientific literature

Rajan, K.[1], Brinkhaus, H.O.[2], Sorokina, M.[3], Zielesny, A.[4], Steinbeck, C.[5]

J. Cheminform. 13, 20 (2021)

DOI: 10.1186/s13321-021-00496-1

This publication has been accepted as a part of Kohulan Rajan's cumulative dissertation (publication equivalence value: 1.0).

Table 2: Author contributions for Publication A

Author No	**1***	**2***	**3**	**4**	**5**
Conceptual research design	x	x	x	x	x
Planning of research activities	x	x		x	x
Tool development (Back-end)	x	x			
Web tool development (Front-end)			x		
Data collection	x	x			
Data analysis and interpretation	x	x			
Manuscript writing	x	x	x	x	x
Suggested publication equivalence value		1.0			
***underlined author numbers refer to involved doctoral students; asterisks mark equal contributions**					

Rajan *et al. J Cheminform* (2021) 13:20
https://doi.org/10.1186/s13321-021-00496-1

Journal of Cheminformatics

SOFTWARE

Open Access

DECIMER-Segmentation: Automated extraction of chemical structure depictions from scientific literature

Kohulan Rajan[1†], Henning Otto Brinkhaus[1†], Maria Sorokina[1], Achim Zielesny[2] and Christoph Steinbeck[1*]

Abstract

Chemistry looks back at many decades of publications on chemical compounds, their structures and properties, in scientific articles. Liberating this knowledge (semi-)automatically and making it available to the world in open-access databases is a current challenge. Apart from mining textual information, Optical Chemical Structure Recognition (OCSR), the translation of an image of a chemical structure into a machine-readable representation, is part of this workflow. As the OCSR process requires an image containing a chemical structure, there is a need for a publicly available tool that automatically recognizes and segments chemical structure depictions from scientific publications. This is especially important for older documents which are only available as scanned pages. Here, we present *DECIMER (Deep lEarning for Chemical IMagE Recognition) Segmentation*, the first open-source, deep learning-based tool for automated recognition and segmentation of chemical structures from the scientific literature. The workflow is divided into two main stages. During the detection step, a deep learning model recognizes chemical structure depictions and creates masks which define their positions on the input page. Subsequently, potentially incomplete masks are expanded in a post-processing workflow. The performance of DECIMER Segmentation has been manually evaluated on three sets of publications from different publishers. The approach operates on bitmap images of journal pages to be applicable also to older articles before the introduction of vector images in PDFs. By making the source code and the trained model publicly available, we hope to contribute to the development of comprehensive chemical data extraction workflows. In order to facilitate access to DECIMER Segmentation, we also developed a web application. The web application, available at https://decimer.ai, lets the user upload a pdf file and retrieve the segmented structure depictions.

Keywords: Deep learning, Image Segmentation, Optical Chemical Structure Recognition, Neural Networks, Chemical data extraction

Introduction

Chemical information is communicated as text and images in scientific publications [1]. These data formats are not intrinsically machine-readable and the manual extraction of chemical information from the literature is a time-consuming and error-prone procedure [2]. Hence, the increasing amount of chemical information being published creates a demand for automated chemical information extraction methods [3].

Over the course of the last three decades, there has been an active development in the field of Optical Chemical Structure Recognition (OCSR). OCSR is the translation of an image of a chemical structure into a machine-readable representation [4]. Most OCSR tools

*Correspondence: christoph.steinbeck@uni-jena.de
†Kohulan Rajan and Henning Otto Brinkhaus contributed equally to the work
[1] Institute for Inorganic and Analytical Chemistry, Friedrich-Schiller-University Jena, Lessingstr. 8, 07743 Jena, Germany
Full list of author information is available at the end of the article

are only capable of processing images with pure chemical structure depictions. Consequently, an automated segmentation of chemical structures from surrounding document information (text, tables etc.) is desirable. Previous approaches for this task are briefly described in the following paragraphs.

The open-source OCSR tool OSRA was published with a rule-based page segmentation algorithm. This mechanism identifies a chemical structure depiction based on the dimensions of a rectangular bounding box around a region of interest and the ratio of black and white pixels within the bounding box [5].

The open-source tool ChemSchematicResolver (CSR) is capable of segmenting images which only contain labels and chemical structure depictions. The classification of objects as labels or structure depictions is done using k-means clustering based on a custom feature density metric. If the publication is available as a markup document, these images can be extracted automatically, so that CSR is capable of processing whole documents [6]. Nevertheless, CSR is incapable of handling scanned pages or images which contain other objects than labels and structure depictions.

In 2019, Staker et al. reported a deep-learning-based OCSR tool which contains a segmentation procedure [7]. Opposed to the previously mentioned feature density-based approaches used by OSRA and CSR, they trained a convolutional neural network based on the U-Net architecture [8] to address the segmentation problem. Every image is processed multiple times at different resolutions and the masks generated by the model are averaged. The model was trained on a semi-synthetic dataset: OSRA was used to identify bounding boxes of potential chemical structure depictions in an unspecified amount of publications and patents. These areas were then cut out of the original documents and replaced with structures from publicly available datasets. During training, the images were randomly modified (with e.g. binarization, brightness adjustments) for data augmentation purposes. The segmentation accuracy has not been reported independently and the accuracy for the whole process of segmentation and structure resolution on different training datasets has been reported to be between 41 and 83% [7]. Unfortunately, the authors have not made their code and the trained models openly available.

With the DECIMER [9] project, we are currently working on the development of an open-source platform for the automated chemical structure extraction from printed literature. It aims at segmenting all chemical structure depictions from a given scanned document from the printed scientific literature and resolving their identity to yield a machine-readable presentation of the molecule. Here, we present *DECIMER Segmentation,* the first step of the DECIMER project and the first openly available deep learning tool for the segmentation of chemical structure depictions from scanned whole-page documents. Perspectively, the segmented chemical structure depictions will be used as input for the DECIMER algorithm, an OCSR method which predicts the SMILES string of the depicted chemical structure.

The algorithm consists of two main stages: First, during the detection step, a deep learning-based model generates masks that define the positions of the chemical structures in a given image. This is followed by a mask expansion procedure during which potentially incomplete masks are expanded until they cover the depictions completely (Fig. 1).

We did not attempt to extract vector graphics from modern PDF articles since this approach would fail for older articles before the early 1990′s, which are mostly scanned pages from printed versions of the journal. Instead, our approach operates on bitmap images of journal pages to be widely applicable also to older articles before the introduction of vector images in PDFs.

The source code of the application described herein as well as the trained model are publically available. Additionally, we created a web application accessible at *decimer.ai* to ensure that the segmentation algorithm becomes widely usable.

Implementation

The DECIMER Segmentation backend mechanism was built using Python 3 with Tensorflow 2.3.0 [10]. It mainly consists of the recognition of chemical structure diagrams using a deep learning model and the subsequent expansion of the resulting masks. The web application is developed in Python 3 using the Django version 3.1.3 framework and React.js for the front-end. The implementation details of the key elements as well as the complete workflow which accepts a pdf document as an input and returns the segmented chemical structure diagrams as an output are described below.

Deep learning algorithm

For the chemical structure detection, a model utilizing the Mask R-CNN network [11] was trained where the Mask R-CNN implementation published by the Matterport team [12] was used with some modifications to work on Tensorflow 2.3.0 with Keras at the backend.

The dataset used for training the model is based on 994 articles from the *Journal of Natural Products* which were chosen arbitrarily from all available issues. We converted the pages of these articles into JPEG images (96 dpi) using the Python pdf2image package [13] and deleted all images that did not contain any chemical structure diagram. After deleting pages which did not contain any

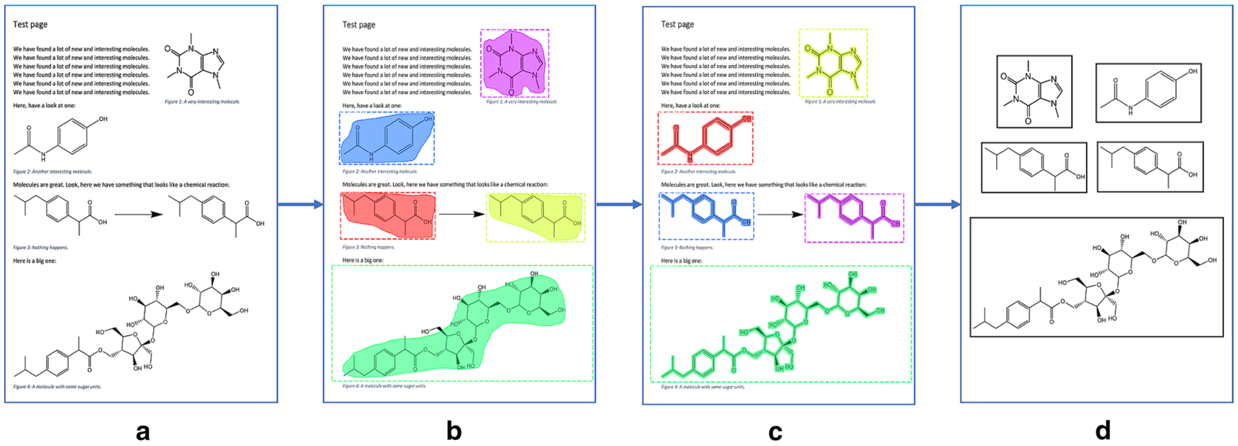


Fig. 1 Graphical summary of the DECIMER Segmentation workflow. The input is an image of a page with chemical structure depictions (**a**). Then, the chemical structure depictions are detected using the Mask-RCNN model (**b**). Subsequently, the masks that define the positions of the depictions are refined and expanded (**c**). Finally, the regions defined by the masks are segmented to yield individual images of chemical structure depictions (**d**)

chemical structure diagrams, there were a total of 1820 pages. The VGG image annotator tool [14] was used to manually annotate the chemical structure diagrams present in each image. Each depiction of a chemical structure was annotated by defining a polygon around it. If there were mechanism arrows or numbers within the structure, these were also included. Other objects like reaction arrows or labels around the chemical structures were not included. This resulted in 9992 annotated regions in the images which each contained one structure diagram (approximately 5.5 annotated structures per image). This dataset was split randomly into a training and validation subset of 90 and 10% respectively.

The model used the hyperparameters pre-defined by the Matterport team, furthermore, we used a batch size of two images per batch, learning rate of 0.001, learning momentum of 0.9, 500 steps per epoch and 50 steps for validation. The model was trained on a compute-server equipped with an Nvidia 1080Ti GPU, 64 GB of RAM and two Intel(R) Xeon(R) Silver 4114 CPUs. The training started from the pre-trained COCO weights provided by the Matterport team. The layers that could not be imported from the pre-trained weights of the model due to different amounts of classes (network heads) were trained for an initial 100 epochs, then the complete model was fine-tuned for another 100 epochs. During the whole training process, the parameters remained the same. This took approximately 26 h in total.

When applying the resulting model to an image of an article page, it returns masks which indicate whether or not a pixel in the original image belongs to a chemical structure diagram. These masks are binary matrices with the first two dimensions of the input image which can contain the values *True* or *False*. This means that every pixel in the original image has a corresponding value in the mask that defines whether or not this pixel is part of a chemical structure depiction. The positional information given in the masks can then be used for the segmentation of the chemical structures.

Mask expansion algorithm

A common problem with the masks generated by the Mask R-CNN model is an unwanted partial coverage of chemical structures only: The model did correctly recognize the chemical structure diagrams on a given page but did not cover them completely (Fig. 2, top row). Therefore, a custom mask expansion algorithm was developed which takes an image and a mask and creates a mask that covers the previously partially detected objects completely.

The expansion workflow begins with the binarization of the input image using a high threshold as recommended by the developers of CSR [6]. The binarization ensures that a non-white background or relicts from low-quality scans are filtered. Then, a binary dilation is applied to turn chemical structure depictions into connected objects, closing, for example, the gaps between an element symbol and its adjacent bonds with non-white pixels. The kernel object used for the dilation is a square with a resolution-dependent size.

Then, the initial seeds for the expansion are determined. For this, the center of the mask is defined as the position in the middle between the highest and the lowest x- and y- coordinates of *True* values of the mask. If the resulting center point is not covered by the mask due to its asymmetric shape, the center point is defined as a

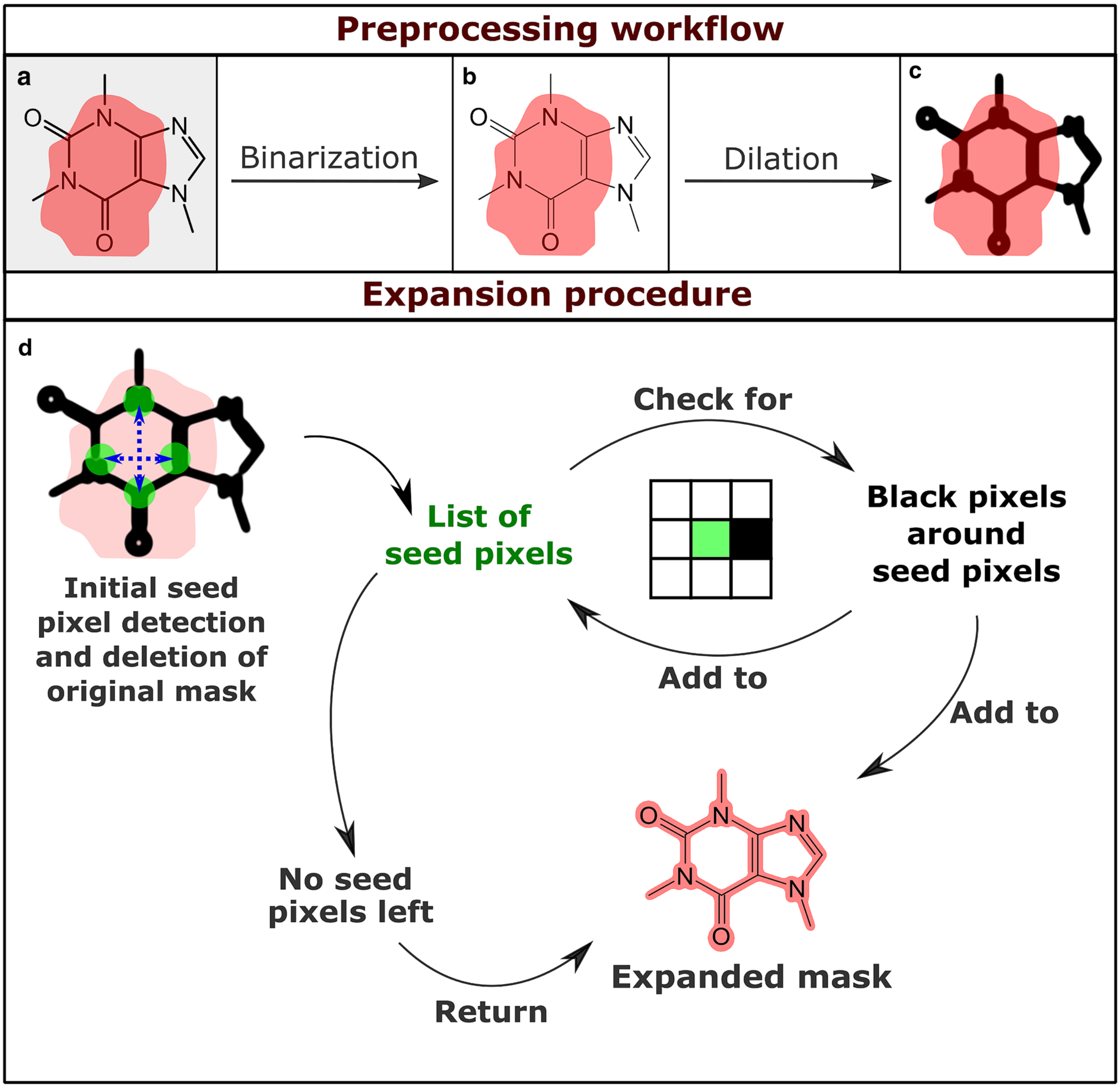


Fig. 2 Mask expansion workflow: During the preprocessing workflow, the original image (**a**) is binarized. The overlaying red patch represents an incomplete mask which is returned by the model. The resulting binary image (**b**) is then dilated to fill gaps within the structure (**c**). This is followed by the expansion procedure (**d**) where the mask is reconstructed by tracing the connected set of non-white pixels starting from a list of seed pixels until no further connected non-white pixels can be found in any direction. This ensures the segmentation of complete chemical structure depictions

random point between the highest and lowest x-coordinates which is covered by the mask. Based on the center point position, the algorithm attempts to determine four black pixels which are covered by the mask in four different directions. If at least one seed pixel is found, the original mask is replaced by a matrix of the same shape which only contains zeros and the expansion is initiated. If no seed pixels have been determined, objects on the contours of the mask are detected as seed pixels. In this case, the original mask is kept and only expanded based on the seed pixels.

The resulting list of seed pixel coordinates is used in the expansion procedure. The eight surrounding pixels of every seed pixel are examined. If one of them is black and not already covered by the mask, the mask is edited to cover it and it is added to the list of seeds. This recursive procedure leads to the inclusion of a complete object in the mask even if the original mask had not covered it completely. This outlined procedure is illustrated in Fig. 2.

The complete tool

DECIMER-Segmentation accepts PDF documents as input and returns grayscale images which contain the segmented chemical structure diagrams. Figure 3 illustrates the workflow.

All pages of the given input PDF document are converted to separate PNG images. All the images are stored in a folder with the name of the input PDF file. During the following procedure, the processing of each image can be parallelized. The structure detection model is initialized for each thread and generates the masks which define the positions of the chemical structure diagrams in the given image. Subsequently, these masks are processed by the expansion algorithm.

The final masks and images are then processed in a segmentation procedure. First, each segment is converted into a grayscale image. Then the maximal width and height of every mask are determined. With this information, an empty image with the dimensions of the resulting segment is created and the chemical structure diagram is placed in it. After all the segments are generated, they are resized into separate square images. These segments are displayed to the user at the end in the web application or saved locally.

Decimer.ai web application

The single-page web application (SPA) is freely available at https://decimer.ai and allows DECIMER usage without any local installation. It is implemented with the Django framework version 3.1.3 to manage the back-end and the API and with the JavaScript React.js library for the front-end. The SPA allows the user to upload a PDF file of a research article, performs image segmentation on it, and returns the extracted molecular images. The latter can be downloaded. The user can also click on the "I'm Feeling Lucky" button, to randomly select a recent article from the Open Access journal *MDPI Molecules* and run the segmentation on it.

Validation

Methods

In order to evaluate the performance of DECIMER Segmentation, we processed 25 articles from the *Journal of Natural Products*, 25 articles from *Phytochemistry* and 25 articles from *Molecules*. None of these journal articles were included in the training dataset. The 75 articles contained a total of 777 pages (365 in *Molecules*, 228 in *Phytochemistry,* 184 in *Journal of Natural Products*) and contained 887 segmented images (398 in *Molecules* 183 in *Phytochemistry*, 306 in *Journal of Natural Products*). We then manually inspected all segmented images to determine if they contain a complete chemical structure diagram or additional objects such as labels or reaction arrows. Furthermore, we determined the number of additional missed structure diagrams on the pages where structures had been determined.

Results and discussion

Without the application of the mask expansion, 81.6% of the segmented images contained complete chemical structure depictions (80.7% in *Molecules*, 83.5% in *Phytochemistry,* 81.7% in *Journal of Natural Products*). Here, a *complete detection* means that the structure was completely covered by the mask. It is necessary to mention that 9.4% (11.1% in *Molecules*, 5.5% in *Phytochemistry,* 9.4% in *Journal of Natural Products*) of these segments contained additional surrounding objects like labels or reaction arrows. Mechanism arrows or atom numbering were not counted as additional objects here as they are often positioned within the structure itself.

When the mask expansion was added to the procedure, the proportion of completely segmented structures increased to 99.8% (99.5% in *Molecules*, 100% in *Phytochemistry,* 100% in *Journal of Natural Products*). Among the validation results there were only two segments which did not contain a chemical structure diagram at all. Unfortunately, the proportion of segments that also contained additional objects rose to 11.2% (12.6% in *Molecules*, 11.5% in *Phytochemistry,* 9.5% in *Journal of Natural Products*). On average, 91.3% of the chemical structures were detected by the model (92.8% in *Molecules*, 86.3% in *Phytochemistry,* 92.7% in *Journal of Natural Products*). These results which represent the final output of DECIMER Segmentation are illustrated in Fig. 4.

Throughout the data used for validation, 885 of the 887 segments contained a complete chemical structure depiction. Given the fact that the model was only trained on articles from *Journal of Natural Products*, it is interesting to note that DECIMER Segmentation performs comparably well on the subsets of *Molecules* and *Phytochemistry* articles. This elucidates the general applicability of DECIMER Segmentation—it is capable of detecting chemical structures in the printed scientific literature in general, independent of specific publisher formats.

The inclusion of additional objects in approximately 11% of the segments is, in many cases, caused by surrounding labels or arrows, which were placed closely to the actual chemical structure diagram by the human creator of the graphic (see Fig. 5). It is worth

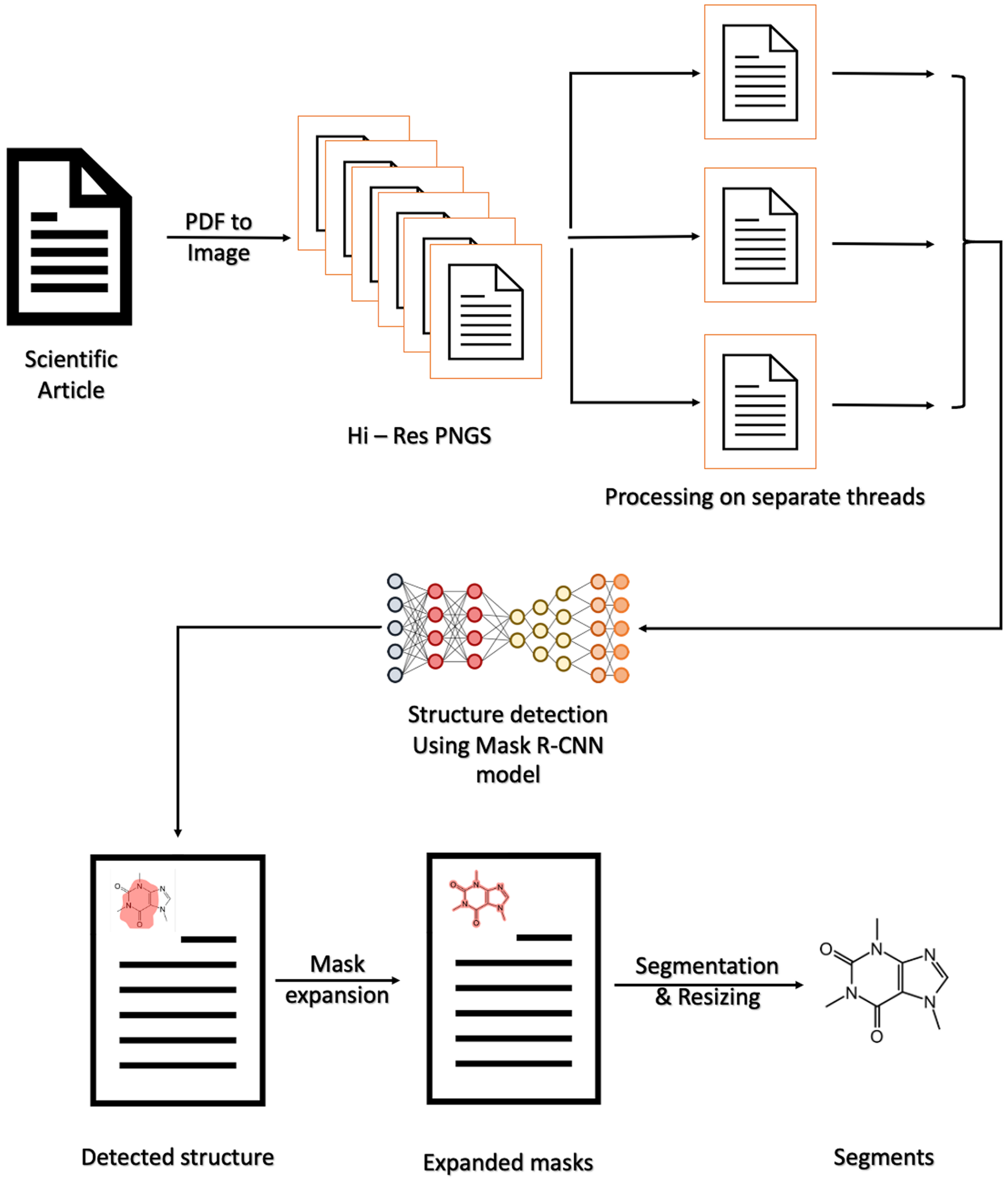


Fig. 3 Visual representation of the complete architecture of the tool

mentioning that the mask expansion sometimes aggravates the problem. For example, in some cases, the tip of a reaction arrow is covered by the mask which is returned by the model. If the arrow is close enough to the structure, the mask expansion leads to its complete inclusion. In other cases, an initially included reaction

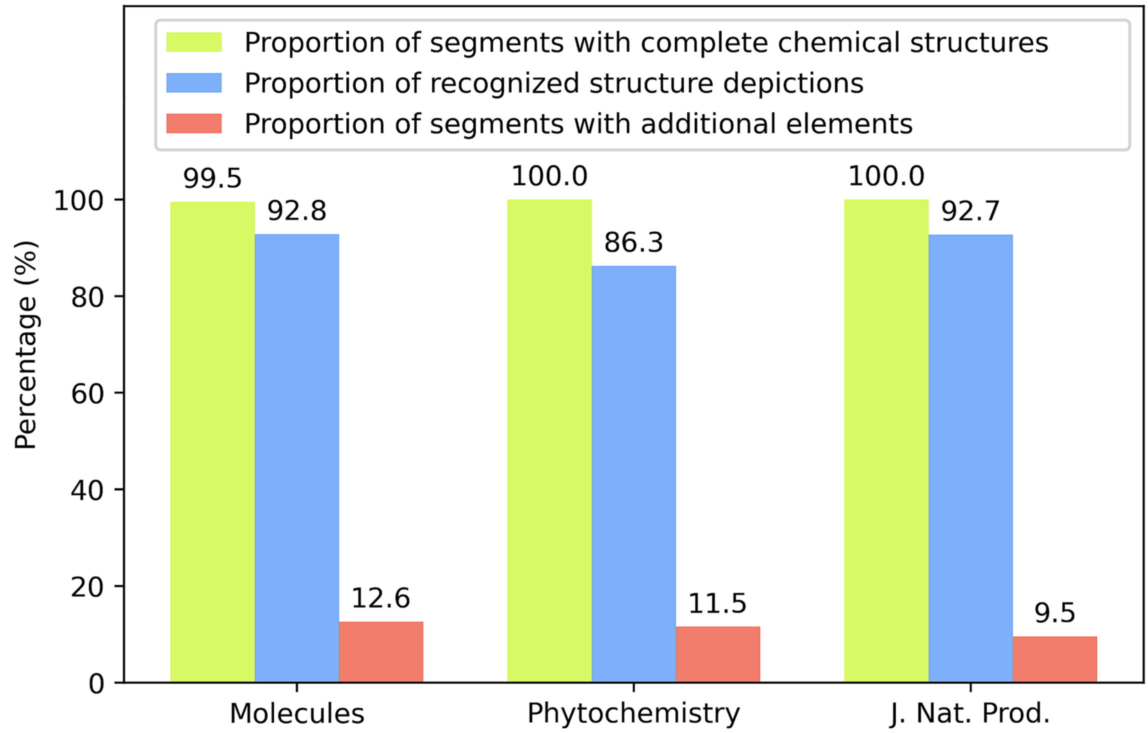


Fig. 4 Overview of the validation results of DECIMER Segmentation

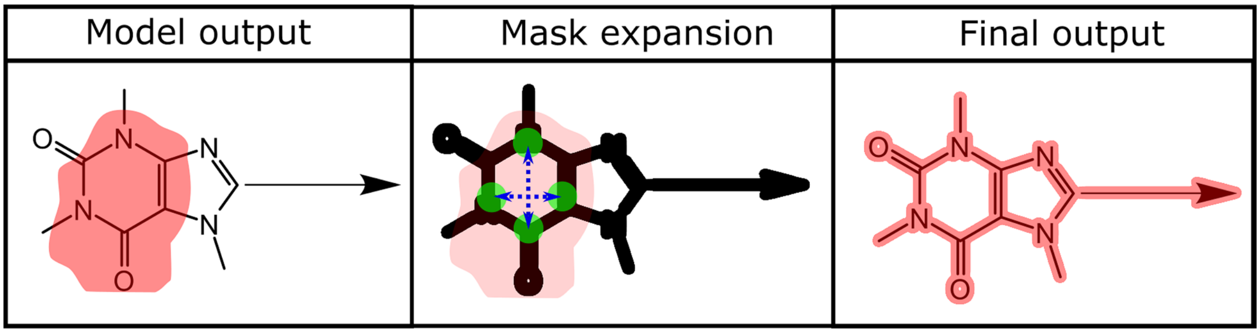


Fig. 5 Exemplary illustration of the wrong inclusion of a reaction arrow in the mask output

arrow may be excluded after the mask expansion if it is not too close to the structure. This is the advantage of choosing seed pixels in the center of the mask which is returned by the model. In an earlier version of DECIMER Segmentation, every pixel of the mask in the model output was included in the final output and the seeds for the expansion were set on the contours of the original mask. This led to wrongly included objects in above 20% of the segments because all objects close to the structure *and* all objects that were wrongly included in the model output were included in the final output. Hence, the mask expansion from seeds in the mask center led to significantly improved results.

During the mask expansion, the application of a binary dilation is necessary to turn the chemical structure depictions into connected objects. This can lead to nearby objects being connected with the structures. This could be addressed by using a smaller kernel for the dilation. The dilemma is that a smaller kernel leads to more cases where the structure is not one connected object which leads to incomplete expansion results. Hence, reducing the amount of wrongly included surrounding objects would necessarily lead to a reduction of the complete segments.

When processing pages parallelly on four threads, on average, the tool took about 1.89 min to process a single article with an average amount of 10.4 pages per article. The time required for processing depends on the number of pages and the number of chemical structures on each page. The numbers mentioned above correspond to an average processing time of 10.9 s per page.

Conclusion

The DECIMER Segmentation tool and the web implementation on decimer.ai for chemical image segmentation are a complete open-source implementation for the segmentation of chemical structure depictions from the published scientific literature.

With the help of deep learning, our method is capable of distinguishing between chemical structures and other content on a page. By applying the system to images, we can mine information from scanned documents which are not available in markup file formats. This allows us to extract information even from the old articles which are only available as scanned files. With the implemented mask expansion process, we are able to segment chemical structure diagrams from the publications completely in high quality.

Although the model was only trained on articles from the *Journal of Natural Products*, we were able to see that the application works well on publications from three different publishers. In future, the detection accuracy of the model can be improved further by training it on an increased amount of publications. In its current state, DECIMER Segmentation can reduce the workload for those who are responsible for the manual creation and curation of chemical databases immensely and could eventually contribute to the full automation of this task.

Abbreviations

API: Application Programming Interface; CNN: Convolutional Neural Network; CPU: Central Processing Unit; CSR: ChemSchematicResolver; DECIMER: Deep lEarning for Chemical IMagE Recognition; GB: Gigabyte; GPU: Graphics Processing Unit; JSON: JavaScript Object Notation; JPEG: Joint Photographic Experts Group; MDPI: Multidisciplinary Digital Publishing Institute; OCSR: Optical Chemical Structure Recognition; OSRA: Optical Structure Recognition Application; PDF: Portable Document Format; PNG: Portable Network Graphics; RAM: Random Access Memory; R-CNN: Region-Based Convolutional Neural Networks; VGG: Visual Geometry Group.

Acknowledgements

The authors would like to thank Ann-Kathrin Brüggemann and Lisa Sophie Kersten for their help in annotating the data.

Authors' contributions

KR and HOB developed the software and performed the analysis, MS developed the web application. CS and AZ conceived the project and supervised the work. All authors contributed to and approved the manuscript.

Funding

Open Access funding enabled and organized by Projekt DEAL. This work was supported by the Carl-Zeiss-Foundation and by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) – Project-ID 239748522 – SFB 1127 ChemBioSys.

Availability of data and materials

The DECIMER—Segmentation web app can be accessed at https://decimer.ai The code for DECIMER—Segmentation is available at https://github.com/Kohulan/DECIMER-Image-Segmentation. Project name: DECIMER. Project home page: https://decimer.ai. Operating system(s): Linux, MacOS and Windows 10. Programming language: Python 3. Other requirements: Python packages: Tensorflow 2.0 or above, pillow, opencv-python, matplotlib, scikit-image, imantics, IPython and pdf2image. License: MIT. Any restrictions to use by non-academics: Not applicable.

Competing interests

AZ is co-founder of GNWI—Gesellschaft für naturwissenschaftliche Informatik mbH, Dortmund, Germany.

Author details

[1] Institute for Inorganic and Analytical Chemistry, Friedrich-Schiller-University Jena, Lessingstr. 8, 07743 Jena, Germany. [2] Institute for Bioinformatics and Chemoinformatics, Westphalian University of Applied Sciences, August-Schmidt-Ring 10, 45665 Recklinghausen, Germany.

Received: 8 January 2021 Accepted: 22 February 2021
Published online: 08 March 2021

References

1. O'Boyle NM, Guha R, Willighagen EL et al (2011) Open data, open source and open standards in chemistry: the Blue Obelisk five years on. J Cheminform 3:1–15
2. Swain MC, Cole JM (2016) ChemDataExtractor: a toolkit for automated extraction of chemical information from the scientific literature. J Chem Inf Model 56:1894–1904
3. Krallinger M, Rabal O, Lourenço A, Oyarzabal J, Valencia A (2017) Information retrieval and text mining technologies for chemistry. Chem Rev 117:7673–7761
4. Rajan K, Brinkhaus HO, Zielesny A, Steinbeck C (2020) A review of optical chemical structure recognition tools. J Cheminform. https://doi.org/10.1186/s13321-020-00465-0
5. Filippov IV, Nicklaus MC (2009) Optical structure recognition software to recover chemical information: OSRA, an open source solution. J Chem Inf Model 49:740–743
6. Beard EJ, Cole JM (2020) ChemSchematicResolver: a toolkit to decode 2d chemical diagrams with labels and R-groups into annotated chemical named entities. J Chem Inf Model 60:2059–2072
7. Staker J, Marshall K, Abel R, McQuaw CM (2019) Molecular Structure extraction from documents using deep learning. J Chem Inf Model 59:1017–1029
8. Ronneberger O, Fischer P, Brox T (2015) U-net: convolutional networks for biomedical image segmentation. Lecture notes in computer science, p 234–241
9. Rajan K, Zielesny A, Steinbeck C (2020) DECIMER: towards deep learning for chemical image recognition. J Cheminform 12:65
10. Abadi M, Agarwal A, Barham P, et al (2016) TensorFlow: large-scale machine learning on heterogeneous distributed systems. arXiv:1603.04467 [cs.DC]
11. He K, Gkioxari G, Dollár P, Girshick R (2017) Mask r-cnn. In: Proceedings of the IEEE international conference on computer vision, p 2961–2969
12. Abdulla W (2017) Mask R-CNN for object detection and instance segmentation on Keras and TensorFlow. https://github.com/matterport/Mask_RCNN. Accessed 1 Dec 2020
13. Belval E. pdf2image. https://github.com/Belval/pdf2image. Accessed 9 Dec 2020
14. Dutta A, Zisserman A (2019) The VIA annotation software for images, audio and video. In: Proceedings of the 27th ACM international conference on multimedia. Association for Computing Machinery, New York, NY, p 2276–2279

Publisher's Note

Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

2.2 Publication B: RanDepict: Random chemical structure depiction generator

Brinkhaus, H.O.[1], Rajan, K.[2], Zielesny, A.[3], Steinbeck, C.[4]

J. Cheminform. 14, 31 (2022)

DOI: 10.1186/10.1186/s13321-022-00609-4

Table 3: Author contributions for Publication B

Author No	1	2	3	4
Conceptual research design	x	x	x	x
Planning of research activities	x	x	x	x
Tool development	x	x		
Data collection	x			
Data analysis and interpretation	x			
Manuscript writing	x	x	x	x
Suggested publication equivalence value	1.0			
*underlined author numbers refer to involved doctoral students				

Journal of Cheminformatics

SOFTWARE Open Access

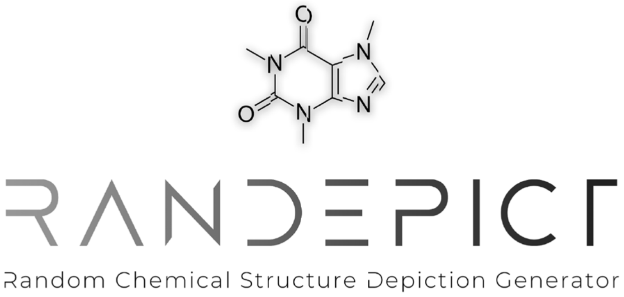
RanDepict: Random chemical structure depiction generator

Henning Otto Brinkhaus[1], Kohulan Rajan[1], Achim Zielesny[2] and Christoph Steinbeck[1*]

Abstract
The development of deep learning-based optical chemical structure recognition (OCSR) systems has led to a need for datasets of chemical structure depictions. The diversity of the features in the training data is an important factor for the generation of deep learning systems that generalise well and are not overfit to a specific type of input. In the case of chemical structure depictions, these features are defined by the depiction parameters such as bond length, line thickness, label font style and many others. Here we present RanDepict, a toolkit for the creation of diverse sets of chemical structure depictions. The diversity of the image features is generated by making use of all available depiction parameters in the depiction functionalities of the CDK, RDKit, and Indigo. Furthermore, there is the option to enhance and augment the image with features such as curved arrows, chemical labels around the structure, or other kinds of distortions. Using depiction feature fingerprints, RanDepict ensures diversely picked image features. Here, the depiction and augmentation features are summarised in binary vectors and the MaxMin algorithm is used to pick diverse samples out of all valid options. By making all resources described herein publicly available, we hope to contribute to the development of deep learning-based OCSR systems.

Keywords: CDK, Chemical image depiction, Depiction generator image augmentation, Indigo, RDKit, OCSR

Graphical Abstract



Introduction

Since 2019, there has been a lot of development in the field of deep learning-based optical chemical structure recognition (OCSR) [1–7]. This indicates a paradigm shift as convolutional neural networks (CNN) as encoders in combination with recurrent neural networks (RNN) or

*Correspondence: christoph.steinbeck@uni-jena.de

[1] Institute for Inorganic and Analytical Chemistry, Friedrich-Schiller-University Jena, Lessingstr. 8, 07743 Jena, Germany
Full list of author information is available at the end of the article

transformers as decoders replace the rule-based systems that have previously defined the standard in the field [8].

The rule-based systems typically apply a workflow of binarisation, vectorisation, the detection of specific structural elements like dashed lines and wedges, optical character recognition (OCR), graph compilation and additional post-processing steps. Every single step in these workflows can be fine-tuned to achieve optimal results. In 2021, Clevert et al. have shown that the openly available rule-based systems surprisingly fail on the common benchmark datasets when slight image perturbations like rotation and shearing are introduced [3]. This lack of robustness is a clear indication that these systems have been overfitted to the benchmark datasets and that there is a need for more diverse benchmark data.

A machine-learning system learns to adapt its actions based on given environment information. Consequently, the quality of the environment information is a crucial factor for the system learning to solve a specific task [9]. Machine-learning systems are able to learn best when the input data they receive is similar to the data they have been trained on. In the case of most deep learning-based OCSR systems, the training data consists of images with depictions of chemical structures which are mapped to string representations of the underlying molecular graph. To be able to generalise well across a variety of different depiction styles, a machine-learning model needs to be trained on these depiction styles as well. Additionally, chemical structure depictions often contain non-structural elements like atom numbering or mechanism arrows which need to be considered as common noise types (Fig. 1). This is particularly relevant for real-world chemical data extraction applications, since the only openly available deep learning-based segmentation tool for chemical structures, DECIMER Segmentation, tends to include these non-structural elements in its output segments [10]. Hence, there is a need for a tool for the generation of chemical structure depictions of various depiction styles with additional non-structural elements.

We present RanDepict, a toolkit for generating diverse representations of chemical structures. It addresses the problem of the generation of diverse training data for OCSR tools by pseudo-randomly setting the available depiction parameters when depicting a structure with one out of three cheminformatics toolkits (Chemistry Development Kit (CDK) [14], RDKit [15] and Indigo [16]). Various augmentations such as image perturbations or non-structural elements like labels and curved arrows can also be added. Instead of pseudo-randomly picking depiction and augmentation parameters, there also is the option to generate the images based on depiction feature fingerprints. Here, the depiction and augmentation parameters are represented as bit arrays and RDKit's implementation of the MaxMin algorithm [17] is used to pick diverse samples out of all valid fingerprints.

By making it publicly accessible, we hope to contribute to the development of robust deep learning-based OCSR systems by providing diverse training and benchmark datasets. RanDepict's source code is publicly available on GitHub.

Implementation

RanDepict is written using Python 3 [18]. The chemical structure depictions are generated using the CDK, RDKit and Indigo. As CDK is Java-based, its classes are accessed in Python via JPype [19].

When a chemical structure depiction is generated, one of the three above-mentioned cheminformatics toolkits is picked randomly. Then, the depiction functions arbitrarily define all available parameters. Among these parameters are bond length, thickness, style, kékulisation, font type and size of atom labels, rotation of molecules, the

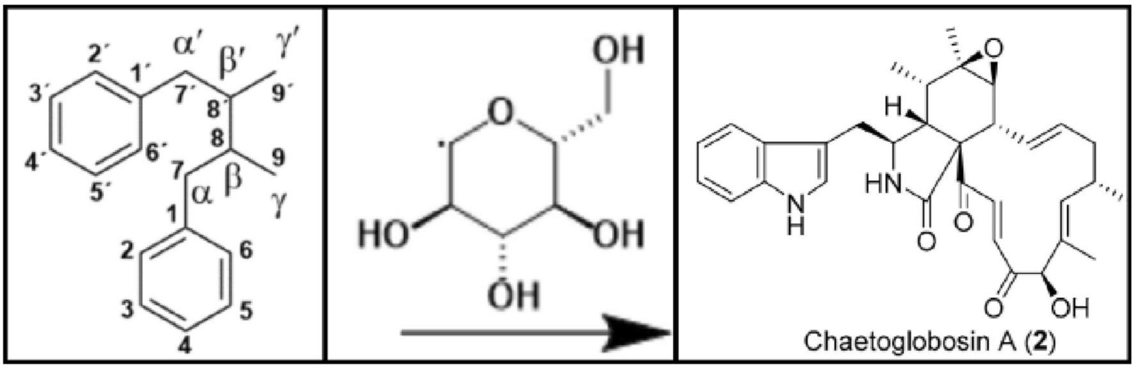


Fig. 1 Examples of structure depictions from chemical publications extracted using DECIMER Segmentation which contain non-structural elements like atom labels (left) [11], reaction arrows (middle) [12] and identity labels (right) [13]

distance between lines and labels and the abbreviation of chemical substructures. Here, the abbreviation of chemical substructures means that, for example, a tertiary butyl group is abbreviated as tBu instead of drawing the full branched chain. Additionally, atom numbering and chirality labels are included in the depiction parameters as they are added by the cheminformatics toolkits and not by separate functions.

Various non-structural features can be added to the structure depiction. Along with atom numbering and chirality labels, there are also curved mechanism arrows, straight reaction arrows, chemical identity labels, rest group labels, and reaction condition labels.

The arrow images are randomly picked from a set of available images, resized, rotated, and pasted in a position where they do (curved arrows) or do not (reaction arrows) overlap with the chemical structure depiction.

The labels are generated by arbitrarily combining a variety of available text elements. For example, a chemical identity label is generated as a number (e.g., '1'), a number-letter combination (e.g., '1a'), a number-number combination (e.g., '1–4') or a number-letter-letter combination ('1a–d'). Similarly, rest group labels are generated by combining rest group variables (e.g., 'R', 'X') with randomly picked superatom labels. The list of superatoms that is used here was originally published along with the rule-based OCSR system OSRA [20]. Reaction condition labels are generated by combining the name of a chemical compound, a solvent, and a time. The font size and type for the labels are randomly chosen. The available font types include standard fonts like Arial and Times New Roman but also fewer common fonts that contain, for example, Asian or Greek-style characters. This ensures that there are diverse types of non-structural elements around the chemical structure that a potential deep learning-based OCSR system can learn to ignore as noise. Furthermore, the image augmentation library imgaug is used to add additional image perturbations. This includes a mild rotation, shearing, salt and pepper noise, brightness and colour adjustments, JPEG compression and pixelation.

Every image created by RanDepict with the desired shape of (m, n) is slightly distorted and resized. Therefore, it is first generated with a shape of (m_{dist}, n_{dist}) where m_{dist} and n_{dist} are randomly drawn from [0.9*m, 1.1*m] and [0.9*n, 1.1*n]. Then, it is resized to the desired shape (m, n) with a randomly picked resizing method. The purpose of this procedure is the introduction of the artefacts of different resizing methods in the image data.

Whenever a (pseudo-)random decision is made, the seed attribute of the RandomDepictor class is used as a seed for the pseudo-random choice and then altered systematically. This ensures that the creation of datasets with RanDepict is reproducible under the condition that the tool is fed the same SMILES input and the same initial seed.

Since the entire depiction parameters constitute a high-dimensional feature space, random sampling does not necessarily guarantee even coverage. Instead of choosing parameters randomly, RanDepict can use depiction feature fingerprints to deal with this issue. This means that all depiction parameters as well as the presence or absence of the different augmentation types are summarised in bit arrays. Here, a 1 or a 0 in every position represents the presence or absence of a certain feature (exemplary illustration in Fig. 2). After computing all possible valid fingerprints, RDKit's implementation of the MaxMin algorithm [17] is used to pick diverse samples. This way, diversity of depiction features is ensured.

The set of all possible valid fingerprints is determined as the combination of all valid fingerprint building blocks in a given order. Here, a fingerprint building block is a valid subset of values that are linked to certain positions in the whole fingerprint which express one depiction feature. A valid fingerprint is a combination of values that does not lead to contradicting statements about the underlying chemical structure depiction.

Let an exemplary chemical structure depiction be defined by the two features kékulisation and bond width. The kékulisation is defined on position 0 of the fingerprint. The resulting building block for this feature is (0, 1) as the first position of the fingerprint can take these two values to refer to whether the kékulisation is being applied or not. Assuming that the bond width can be thin, medium, or bold, these options would be described by positions 1–3 of the fingerprint. The building blocks for the feature bond width would be (1,0,0), (0,1,0) and (0,0,1). Other combinations for these positions would be invalid as, for example, the combination (1,0,1) on these fingerprint positions would refer to the bond width being thin and bold at the same time. The combination of the valid building blocks for all features in the given order defines the set of all fingerprint combinations. In the aforementioned example, this results in (0,1,0,0), (0,0,1,0), (0,0,0,1), (1,1,0,0), (1,0,1,0) and (1,0,0,1) as the set of valid fingerprints.

The building blocks of the fingerprints are generated automatically. A pseudo-random decision during the depiction creation just needs to be flagged as relevant for the fingerprint. RanDepict recognises this and automatically generates a fingerprint scheme. This way, the code for the fingerprint generation does not need to be adapted in the case of modifications in the depiction creation process.

During the fingerprint generation process, every binary decision (kékulisation in the example above) is simply

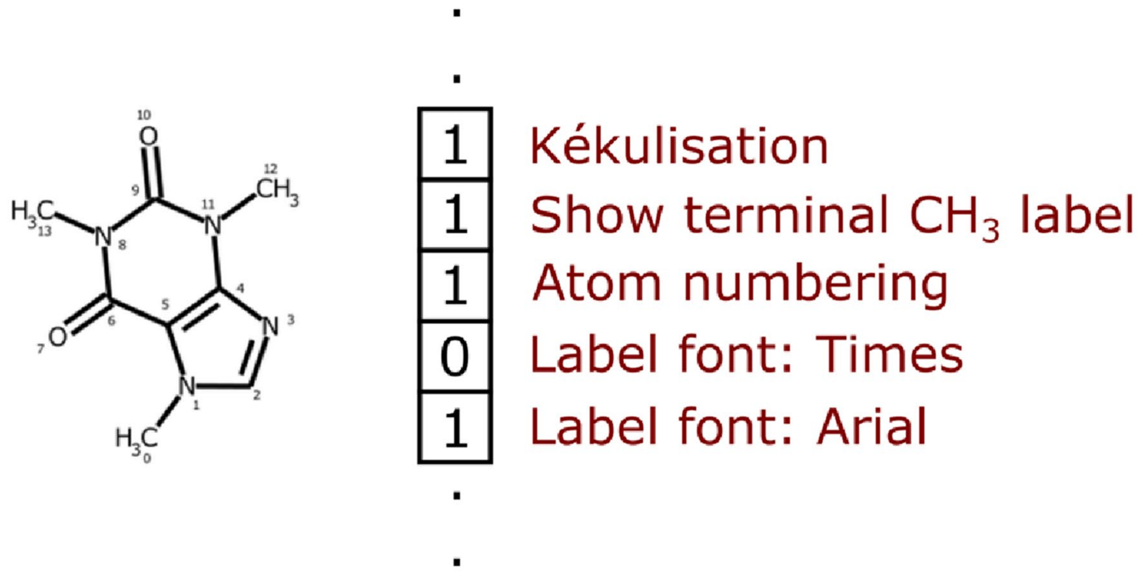


Fig. 2 Exemplary illustration of depiction feature fingerprints

allocated to one position in the bit array. When categorical decisions (bond width in the example above) are allocated to as many positions as there are categories where every position then indicates the presence or absence of a certain category and only one of them can have the value 1. Numerical ranges are split into three subranges which are then treated like categories. For example, if the bond width could be described by an integer with the possible values [1, 2, 3, 4, 5, 6] this would be allocated to three positions in the fingerprint. These positions would be linked to the subsets [1, 2], [3, 4] and [5, 6]. This means that the fingerprint does not always define an exact value for certain parameters but only specifies a range. When creating a depiction from a fingerprint, the parameter is randomly drawn from this subrange. This is necessary to reduce the number of possible fingerprints as the combinatorial explosion complicates computing all possible fingerprint combinations otherwise.

The three cheminformatics toolkits offer varying amounts of adjustable parameters. During the creation of a CDK depiction, 15 parameters are set. When using RDKit and Indigo, 10 and 8 parameters are adjustable. The ranges of possible values for these parameters differ between the tools. Hence, fingerprints for CDK, RDKit and Indigo depictions and the additional augmentations are four separate entities. The augmentation fingerprints only describe the presence or absence of an augmentation feature but do not comprise the specific parameters which are set. The varying parameter numbers and ranges lead to strongly differing numbers of valid depiction feature fingerprints: 2,799,360 for the CDK fingerprints, 18,432 for RDKit fingerprints, 864 for Indigo and 2048 for the augmentations. When generating a dataset from the fingerprints the user can specify the desired proportions of CDK, RDKit and Indigo depictions as well as the proportion of structures with added augmentations. They default to 55% (CDK), 30% (RDKit) and 15% (Indigo), 50% (augmented).

Results

RanDepict was designed to allow the generation of diverse chemical structure depictions using only a few lines of code. After generating a RandomDepictor object, the method random_depiction can be used to generate depictions of chemical structures. These depictions are generated by using randomly picked parameters in CDK, RDKit and Indigo without additional elements (Fig. 3). The object can be called as a function to generate chemical structure depictions with additional non-structural elements and augmentations (Fig. 4). There are various examples for the batch generation of structure depiction datasets with and without the usage of the feature fingerprint picking functionality in the documentation.

```
from RanDepict import RandomDepictor

smiles = "CN1C = NC2 = C1C(= O)N(C(= O)N2C)C"

with RandomDepictor() as depictor:
    # Generate chemical structure depictions
    image = depictor.random_depiction(smiles)

    # Generate augmented chemical structure depictions
    augmented_image = depictor(smiles)
```

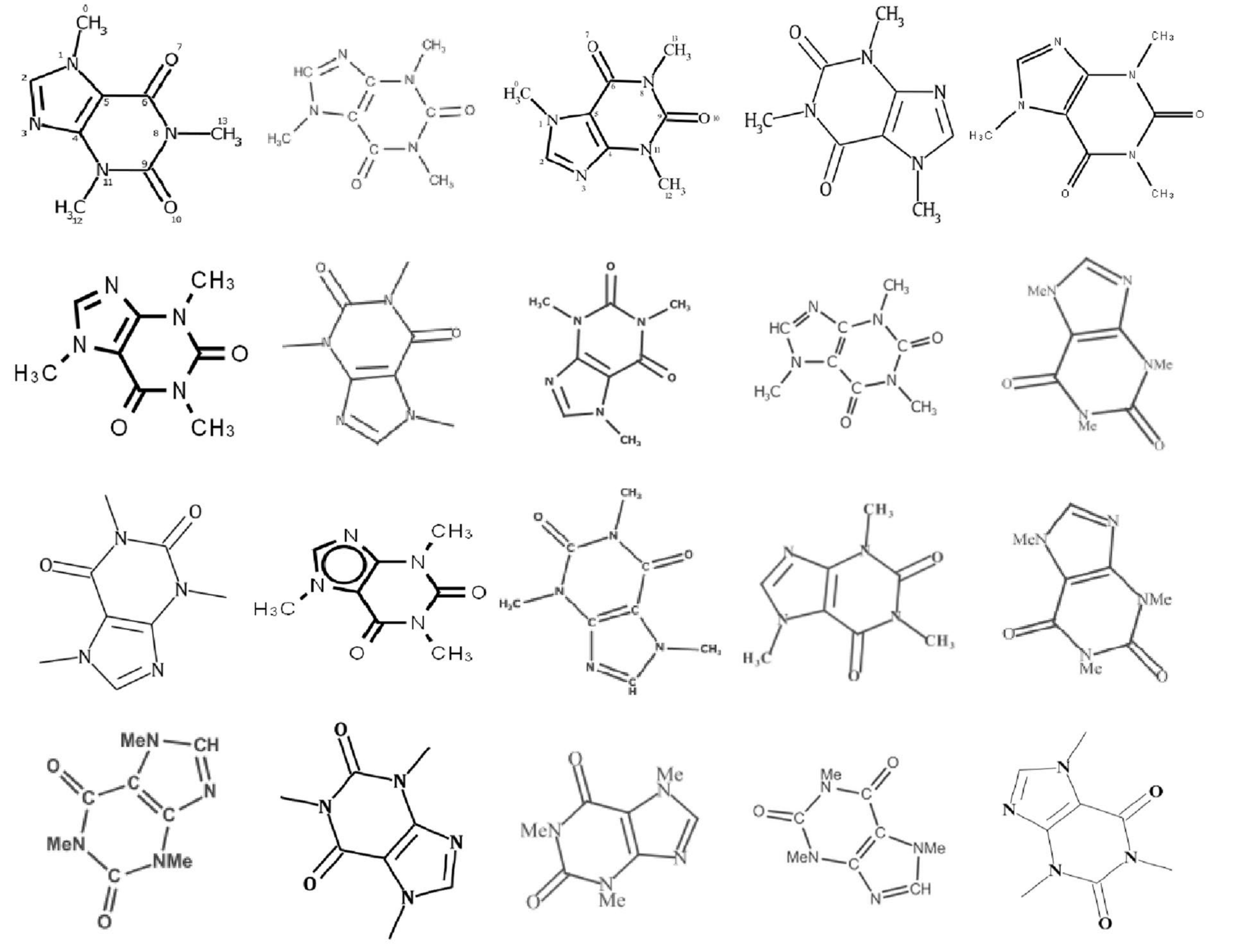


Fig. 3 Depictions of caffeine with various depiction styles generated with RanDepict with feature fingerprint picking without additional augmentations

On a compute server with two Intel(R) Xeon(R) Silver 4114 CPUs and 64 GB of RAM, the runtime was evaluated for the generation of 100, 200, 400, 800, 1600, 3200 and 6400 chemical structure depictions with an image size of 299 × 299 (Fig. 5) using one CPU core. This was done with and without the addition of augmentations and the usage of the feature fingerprints. The linear regression results of the different runs clearly indicate that the runtime increases linearly with a growing amount of depictions.

Based on the regression analysis, the generation of one million chemical structure depictions without the feature fingerprints takes 19 h without augmentations and 31 h with augmentations. For the generation of large datasets consisting of millions of structures, it is recommended to split the input SMILES lists and run the generation in parallel on multiple nodes in a cluster or using a cloud service. As long as the initial seed is set differently in every parallel instance, different sets of parameters are picked.

The same extrapolation applied to the generation of one million structures using feature fingerprint selection results in 127 h without augmentations and 138 h with augmentations. The user could split up the input SMILES lists here, too, and initialise the MaxMin picking mechanism with different seeds on every instance in a computing cluster to ensure different sets of parameters are picked. Nevertheless, the creation of datasets from fingerprints is significantly slower than the generation with random parameter sampling. Depending on the desired dataset size, the user can decide whether to use depiction feature fingerprints. The feature fingerprint picking functionality is highly recommended for the generation of smaller test and benchmark sets as it ensures a diverse selection of features.

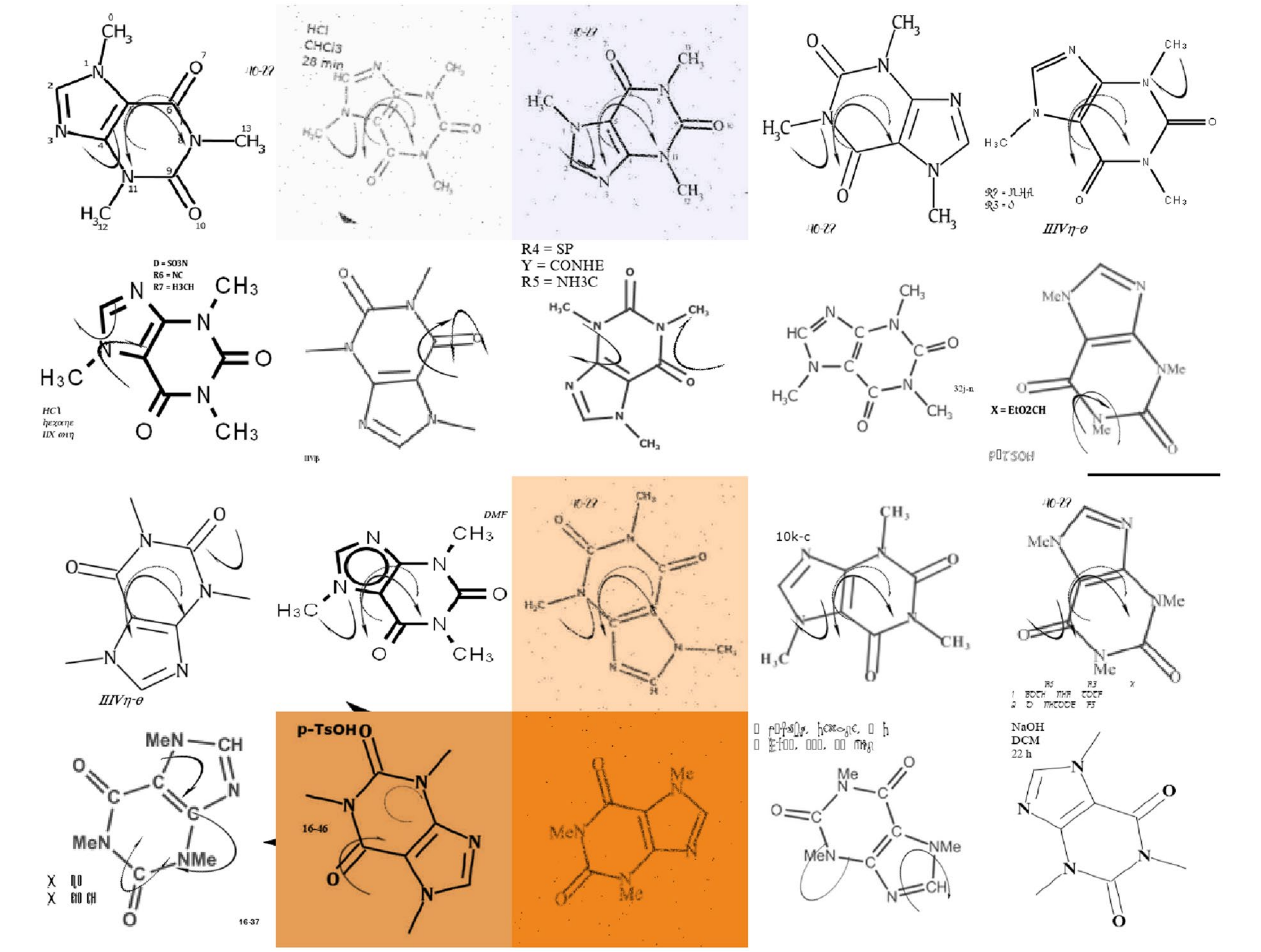


Fig. 4 Depictions of caffeine with various depiction styles and additional non-structural features and noise types generated with RanDepict using feature fingerprint picking

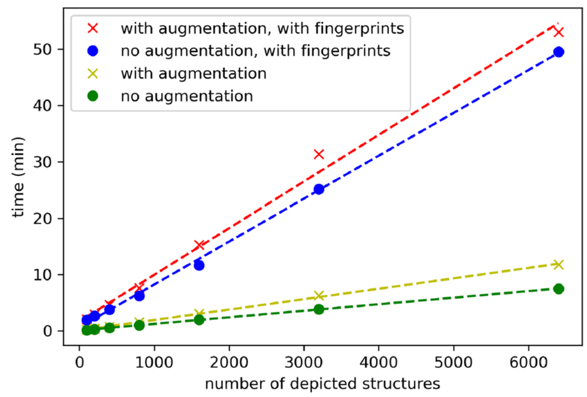


Fig. 5 Runtime analysis of chemical structure depiction generation with RanDepict with and without augmentations and the application of the feature fingerprint picking functionality. The dotted lines represent linear regression results for each case

Conclusions

RanDepict: a toolkit for generating chemical structure depictions. It features diverse structure depiction elements, as well as non-structural elements and image augmentations.

If desired, the diversity of depiction features is ensured by representing the entirety of features in bit arrays (feature fingerprints) and picking diverse sets using the MaxMin algorithm. Even though fingerprint picking is a time-consuming process, we highly recommend using it for the generation of smaller test sets where the random sampling of depiction features may not necessarily lead to a dataset that represents the entire feature space.

The complete source code of RanDepict, scripts for the generation of Figs. 3 and 4, the runtime determination as well as other examples for the usage and detailed documentation of RanDepict are openly accessible on GitHub and Read the Docs. It is possible to install RanDepict as a package via pip. We hope that our work will contribute to the standardisation of training and test datasets in the field of OCSR.

Abbreviations
CDK: Chemistry development kit; CNN: Convolutional neural network; JPEG: Joint Photographic Experts Group; OCSR: Optical chemical structure recognition; OSRA: Optical structure recognition application; RNN: Recurrent neural network; SMILES: Simplified molecular input line entry specification.

Author contributions
HOB developed the Python software and performed the analysis, KR and HOB initiated, designed, tested, applied, and validated the application features. CS and AZ conceived the project and supervised the work. All authors read and approved the final manuscript.

Funding
Open Access funding enabled and organized by Projekt DEAL. This work was supported by the Carl-Zeiss-Foundation and by the German Research Foundation within the framework CRC1127 ChemBioSys.

Availability of data and materials
Project name: RanDepict. Project home page: https://github.com/OBrink/RanDepict, https://pypi.org/project/RanDepict/. Operating system(s): Linux, macOS and Windows 10. Programming language: Python 3. Other requirements: Python packages: numpy >= 1.19, imgaug, scikit-image, epam. indigo, jpype1, ipyplot, rdkit-pypi, imagecorruptions, pillow >= 8.2.0; Java Libraries: CDK 2.5. License: MIT. Any restrictions to use by non-academics: Not applicable.

Declarations

Competing interests
AZ is co-founder of GNWI—Gesellschaft für Naturwissenschaftliche Informatik mbH, Dortmund, Germany.

Author details
[1]Institute for Inorganic and Analytical Chemistry, Friedrich-Schiller-University Jena, Lessingstr. 8, 07743 Jena, Germany. [2]Institute for Bioinformatics and Chemoinformatics, Westphalian University of Applied Sciences, August-Schmidt-Ring 10, D-45665 Recklinghausen, Germany.

Received: 28 February 2022 Accepted: 9 May 2022
Published online: 06 June 2022

References

1. Oldenhof M, Arany A, Moreau Y, Simm J (2020) ChemGrapher: optical graph recognition of chemical compounds by deep learning. J Chem Inf Model 60:4506–4517
2. Khokhlov I, Krasnov L, Fedorov M, Sosnin S (2022) Image2SMILES: transformer-based molecular optical recognition engine. Chem Methods. https://doi.org/10.1002/cmtd.202100069
3. Clevert D-A, Le T, Winter R, Montanari F (2021) Img2Mol - accurate SMILES recognition from molecular graphical depictions. Chem Sci 12:14174–14181
4. Rajan K, Zielesny A, Steinbeck C (2021) DECIMER 1.0: deep learning for chemical image recognition using transformers. J Cheminform 13:61
5. Rajan K, Zielesny A, Steinbeck C (2020) DECIMER: towards deep learning for chemical image recognition. J Cheminform 12:65
6. Weir H, Thompson K, Woodward A, Choi B, Braun A, Martínez TJ (2021) ChemPix: automated recognition of hand-drawn hydrocarbon structures using deep learning. Chem Sci 12:10622–10633
7. Staker J, Marshall K, Abel R, McQuaw CM (2019) Molecular structure extraction from documents using deep learning. J Chem Inf Model 59:1017–1029
8. Rajan K, Brinkhaus HO, Zielesny A, Steinbeck C (2020) A review of optical chemical structure recognition tools. J Cheminform 12:60
9. Wang H, Ma C, Zhou L (2009) A brief review of machine learning and its application. 2009 Int Conf Inf Eng Comput Sci. https://doi.org/10.1109/iciecs.2009.5362936
10. Rajan K, Brinkhaus HO, Sorokina M, Zielesny A, Steinbeck C (2021) DECIMER-Segmentation: automated extraction of chemical structure depictions from scientific literature. J Cheminform 13:20
11. Runeberg PA, Agustin D, Eklund PC (2020) Formation of tetrahydrofurano-, aryltetralin, and butyrolactone norlignans through the epoxidation of 9-norlignans. Molecules. https://doi.org/10.3390/molecules25051160
12. Zhang G, Li Y, Wei W, Li J, Li H, Huang Y, Guo D-A (2020) Metabolomics combined with multivariate statistical analysis for screening of chemical markers between andgentiana scabra and gentiana rigescens. Molecules. https://doi.org/10.3390/molecules25051228
13. Luo X-W, Gao C-H, Lu H-M, Wang J-M, Su Z-Q, Tao H-M, Zhou X-F, Yang B, Liu Y-H (2020) HPLC-DAD-guided isolation of diversified chaetoglobosins from the coral-associated fungus C2F17. Molecules. https://doi.org/10.3390/molecules25051237
14. Steinbeck C, Han Y, Kuhn S, Horlacher O, Luttmann E, Willighagen E (2003) The chemistry development kit (CDK): an open-source Java library for chemo- and bioinformatics. J Chem Inf Comput Sci 43:493–500
15. RDKit: Open-source cheminformatics. https://www.rdkit.org/. Accessed 16 May 2022
16. Indigo Toolkit. https://lifescience.opensource.epam.com/indigo/. Accessed 25 Jun 2020
17. Ashton M, Barnard J, Casset F, Charlton M, Downs G, Gorse D, Holliday J, Lahana R, Willett P (2002) Identification of diverse database subsets using property-based and fragment-based molecular descriptions. Quant Struct Act Relatsh 21:598–604
18. Van RG, Drake F (2009) Python 3 reference manual. CreateSpace, Scotts Valley
19. Nelson KE, Scherer MK, Others (2020) JPype. Lawrence Livermore National Lab (LLNL), Livermore
20. Filippov IV, Nicklaus MC (2009) Optical structure recognition software to recover chemical information: OSRA, an open source solution. J Chem Inf Model 49:740–743

Publisher's note

Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

2.3 Publication C: DECIMER - hand-drawn molecule images dataset

Brinkhaus, H.O.[1], Zielesny, A.[2], Steinbeck, C.[3], Rajan, K.[4]

J. Cheminform. 14, 36 (2022)

DOI: 10.1186/s13321-022-00620-9

Table 4: Author contributions for Publication C

Author No	**1**	**2**	**3**	**4**
Conceptual research design	x	x	x	x
Planning of research activities	x	x	x	x
Data collection	x	x		x
Data analysis and curation	x			x
Manuscript writing	x	x	x	x
Suggested publication equivalence value	1.0			
***underlined author numbers refer to involved doctoral students**				

Brinkhaus *et al. Journal of Cheminformatics* (2022) 14:36
https://doi.org/10.1186/s13321-022-00620-9

Journal of Cheminformatics

DATA NOTE

Open Access

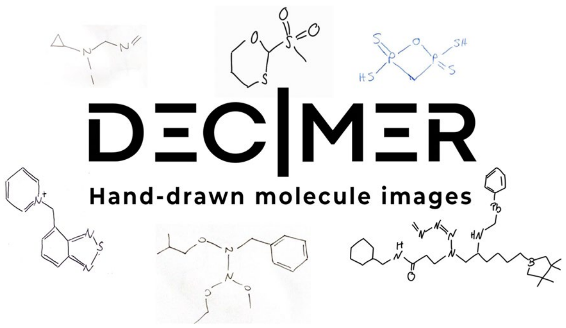
DECIMER—hand-drawn molecule images dataset

Henning Otto Brinkhaus[1], Achim Zielesny[2], Christoph Steinbeck[1] and Kohulan Rajan[1*]

Abstract

The translation of images of chemical structures into machine-readable representations of the depicted molecules is known as optical chemical structure recognition (OCSR). There has been a lot of progress over the last three decades in this field, but the development of systems for the recognition of complex hand-drawn structure depictions is still at the beginning. Currently, there is no data for the systematic evaluation of OCSR methods on hand-drawn structures available. Here we present *DECIMER — Hand-drawn molecule images*, a standardised, openly available benchmark dataset of 5088 hand-drawn depictions of diversely picked chemical structures. Every structure depiction in the dataset is mapped to a machine-readable representation of the underlying molecule. The dataset is openly available and published under the CC-BY 4.0 licence which applies very few limitations. We hope that it will contribute to the further development of the field.

Graphical Abstract



Objective

Most chemical information is published in text and images in the primary scientific literature. The automated conversion of these unstructured, human-readable data formats into structured, machine-readable representations is essential to make the information available in publicly accessible databases. The reliable extraction of information from the depictions of the chemical structures is an ongoing challenge that still has not been fully solved yet. Chemical structure depictions are converted into computer-readable representations using optical chemical structure recognition (OCSR) systems [1].

The field of OCSR has developed significantly over the last 30 years. Most OCSR tools follow a hard-coded set of rules to assemble the underlying molecule based on the

*Correspondence: kohulan.rajan@uni-jena.de

[1] Institute for Inorganic and Analytical Chemistry, Friedrich-Schiller-University Jena, Lessingstr. 8, 07743 Jena, Germany
Full list of author information is available at the end of the article

elements in the vectorised image [2–11]. By 2020 several deep learning-based solutions are available [12–18].

In order to evaluate the performance of the available OCSR tools, realistic benchmark datasets are necessary. At present, there are four real-world datasets available [1, 9, 19] that contain chemical structure depictions that were collected and curated from publications and patents. The evaluation of the performance on realistic data is crucial to demonstrate whether the tools are robust enough to be used in an automated chemical literature mining process.

The resolution of hand-drawn chemical structures is a more challenging task than the resolution of automatically generated depictions. In addition to the varying depiction features which are present anyway, the individual, unique way of drawing the structure adds an increased level of complexity. In 2021, the deep learning-based OCSR tool ChemPix [15] demonstrated its capability to interpret simple hand-drawn hydrocarbon structures with high accuracy. There also are a few closed-source methods and commercial systems available that claim to be capable of resolving hand-drawn chemical structures [20–22]. The authors of the deep-learning-based OCSR tool img2mol demonstrated the capability of their tool to recognise some hand-drawn chemical structures that they had picked themselves and noted the lack of a standardised benchmark set [14].

With the development of more OCSR tools that focus on the resolution of hand-drawn chemical structure depictions, there is a need for a standardised dataset to evaluate their performance. Here we present *DECIMER — Hand-drawn molecule images*, a set of 5088 hand-drawn chemical structures depictions. Every image is mapped to a machine-readable representation of the underlying molecule. The diversely picked molecules represent a wide variety of small molecules. The dataset was created to facilitate the ongoing development in the field of OCSR and is openly accessible.

Data description

The dataset consists of 5088 PNG images of unique hand-drawn chemical structure depictions (Fig. 1) which are mapped to their corresponding SMILES [23] string as well as an SD file. The structures have been drawn by 24 volunteers from the Westphalian University of Applied Sciences, Campus Recklinghausen, Germany, who have graciously offered to use their free time to contribute to the generation of this dataset.

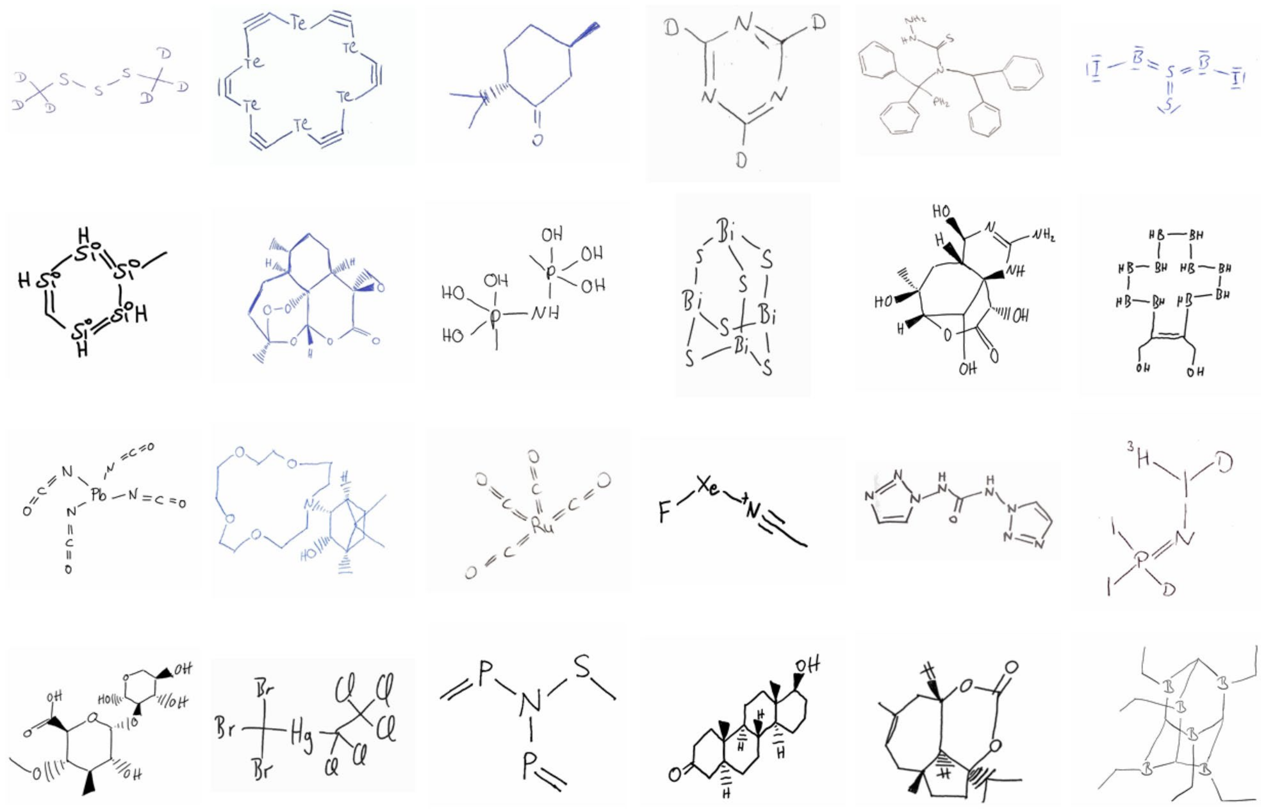


Fig. 1 Examples of hand-drawn chemical structure depictions from the dataset

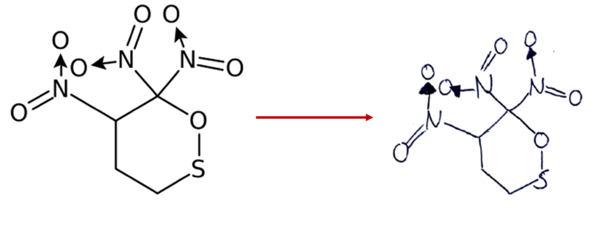

ID: CDK_Depict_46_36

Fig. 2 A chemical structure depiction generated by CDK, sketched on a sheet of paper and scanned as an image file

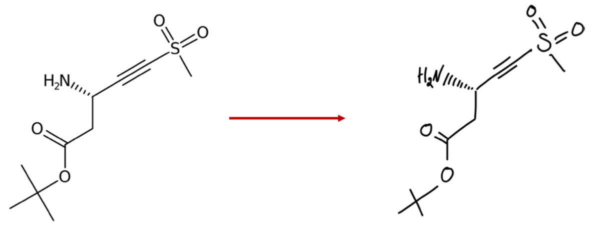

ID: CDK_Depict_45_18

Fig. 3 A chemical structure depiction generated by CDK, sketched on a tablet and saved as an image file

The molecules have been picked from all structures in PubChem [24] using RDKit's implementation of the MaxMin algorithm [25] based on Morgan fingerprints [26] to ensure a diverse coverage of the chemical space. The only filtering rule that has been applied is a molecular weight maximum of 1500 Da. As a consequence, features like stereochemical information, charged groups as well as different types of isotopes are present in the dataset.

There are two categories of images:

Drawn on a piece of white paper and scanned (Fig. 2)
Drawn using a mobile device or tablet and directly saved as an image (Fig. 3).

Curation

In total, 6000 diverse molecules were selected from PubChem using RDKit's implementation of the MaxMin algorithm based on Morgan fingerprints. Subsequently, CDK Depict [27], a structure depiction generator based on the Chemistry Development Kit (CDK) [28], was used to create production-quality 2D images in batches. Each batch of images was then converted into PDF files and they were distributed among the volunteers. Using the chemical structure depictions generated by CDK as a visual template, each volunteer drew the structures on a piece of paper using a black or blue pen or on their tablet using an input device.

Each volunteer sent back the scanned images or the images generated using their device after completing a batch. The curators reviewed the drawings, manually confirmed the correctness of the molecules, cropped the scanned images and stored them in separate image files. As part of the curation, structures that weren't correct due to human error were discarded. A total of 568 images out of 6000 were rejected due to issues with the depicted structure. Another 344 structures were not returned by the volunteers. This resulted in the final dataset of 5088 images in total.

An identifier was assigned to each image, and the same identifier was used to label the SD file which was generated using the CDK. Additionally, the dataset contains a file containing a table of the identifiers and corresponding SMILES representations.

FAIR-ification

The following steps were taken in order to make the dataset findable, accessible, interoperable and reusable (FAIR) [29]. The dataset was deposited in a publicly accessible data repository, in this case, Zenodo. This ensures that the dataset is easily findable. Furthermore, Zenodo provides a digital object identifier (DOI) that can be used to locate the dataset and it can also easily be integrated into Github as well. With Zenodo being an open, public repository, the dataset can be accessed from any part of the globe. To make it as interoperable as possible, the generated images use PNG as the final image format, which can be used across a variety of operating systems. Additionally, SMILES and SDF are representations of chemical structures which can be read by every cheminformatics toolkit. The dataset has been published under the CC-BY 4.0 licence. This licence includes that every user can redistribute or change the data as much as they want as long as they refer to the original authors when publishing results based on it. It is possible to use the data for non-commercial or commercial purposes without further obligations.

Limitation

No restrictions or limitations apply to using and reusing the dataset. Everyone can use this dataset as a standard benchmark set for the evaluation of the performance of their OCSR tools. The dataset includes a wide range of chemical structures and represents a much larger chemical space. The structures were drawn by various individuals to ensure the diversity of drawing styles. The main limitation is caused by the molecular weight filter (< 1500Da) as it excludes certain molecules like big macrocycles, proteins or artificial polymers. Additionally, Markush structures are not represented.

Due to the limited number of images in this dataset, we do not recommend attempting to train a deep learning model using this dataset. We highly recommended using it exclusively for benchmarking instead of fitting the tools to the dataset.

Abbreviations

CDK: Chemistry development kit; CC: Creative commons; DOI: Digital object identifier; FAIR: Findable, accessible, interoperable, and reusable; OCSR: Optical chemical structure recognition; PDF: Portable document format; PNG: Portable network graphics; SDF: Structural data file; SDG: Structure diagram generator; SMILES: Simplified molecular-input line-entry system.

Acknowledgements

The authors would like to acknowledge the following students of the Westphalian University of Applied Sciences for their contribution to the project:Muammer Ates, Samuel Behr, Thilo Bredtmann, Michele Cassano, Saaruky Chanthirakanthan, Zeynep Dagtekin, Dustin Finke, Janina Gleißenberger, Safiye Sarah Kantar, Irina Krützner, Lilly Maddox, Fabian Matten, Timo van Meegdenburg, Alisa Muminovic, Paulina Paszko, Josefine Reiser, Maximilian Rottmann, Lisa Scharfenberg, Patricia Schoof, Betül Sevindik, Fatma Zehra Sevindik, Kaan Sipahi and Julia Zielinski.

Author contributions

KR and HOB initiated the project and curated the final dataset. CS and AZ conceived the project and supervised the work. All authors read and approved the final manuscript.

Funding

Open Access funding enabled and organized by Projekt DEAL. This work was supported by the Carl-Zeiss-Foundation and by the German Research Foundation within the framework CRC1127 ChemBioSys.

Availability of data and materials

The dataset is openly available at ZENODO: https://doi.org/10.5281/zenodo.6456306.

Declarations

Competing interests

AZ is co-founder of GNWI—Gesellschaft für naturwissenschaftliche Informatik mbH, Dortmund, Germany.

Author details

[1]Institute for Inorganic and Analytical Chemistry, Friedrich-Schiller-University Jena, Lessingstr. 8, 07743 Jena, Germany. [2]Institute for Bioinformatics and Chemoinformatics, Westphalian University of Applied Sciences, August-Schmidt-Ring 10, 45665 Recklinghausen, Germany.

Received: 20 April 2022 Accepted: 25 May 2022
Published: 9 June 2022

References

1. Rajan K, Brinkhaus HO, Zielesny A, Steinbeck C (2020) A review of optical chemical structure recognition tools. J Cheminform 12:60 [cito:cites] [cito:citesAsAuthority]
2. McDaniel JR, Balmuth JR (1992) Kekule: OCR-optical chemical (structure) recognition. J Chem Inf Comput Sci 32:373–378 [cito:cites]
3. Casey R, Boyer S, Healey P, Miller A, Oudot B, Zilles K (1993) Optical recognition of chemical graphics. In: Proceedings of 2nd International Conference on Document Analysis and Recognition (ICDAR '93), pp 627–631 [cito:cites]
4. Ibison P, Jacquot M, Kam F, Neville AG, Simpson RW, Tonnelier C, Venczel T, Johnson AP (1993) Chemical literature data extraction: the CLiDE project. J Chem Inf Comput Sci 33:338–344 [cito:cites]
5. Valko AT, Johnson AP (2009) CLiDE Pro: the latest generation of CLiDE, a tool for optical chemical structure recognition. J Chem Inf Model 49:780–787 [cito:cites]
6. Zimmermann M (2011) Chemical structure reconstruction with chemoCR. In: The Twentieth Text REtrieval conference (TREC 2011) Proceedings [cito:cites]
7. Filippov IV, Nicklaus MC (2009) Optical structure recognition software to recover chemical information: OSRA, an open-source solution. J Chem Inf Model 49:740–743 [cito:cites]
8. Park J, Rosania GR, Shedden KA, Nguyen M, Lyu N, Saitou K (2009) Automated extraction of chemical structure information from digital raster images. Chem Cent J 3:4 [cito:cites]
9. Sadawi N (2009) Recognising chemical formulas from molecule depictions. In: Pre-proceedings of the 8th IAPR international workshop on graphics recognition (GREC 2009). pp 167–175 [cito:cites]
10. Tharatipyakul A, Numnark S, Wichadakul D, Ingsriswang S (2012) ChemEx: information extraction system for chemical data curation. BMC Bioinformatics 13(Suppl 17):S9 [cito:cites]
11. Beard EJ, Cole JM (2020) Chemschematicresolver: a toolkit to decode 2D chemical diagrams with labels and R-groups into annotated chemical named entities. J Chem Inf Model 60:2059–2072 [cito:cites]
12. Rajan K, Zielesny A, Steinbeck C (2021) DECIMER 1.0: deep learning for chemical image recognition using transformers. J Cheminform 13:61 [cito:cites] [cito:citesAsAuthority] [cito:extends]
13. Rajan K, Zielesny A, Steinbeck C (2020) DECIMER: towards deep learning for chemical image recognition. J Cheminform 12:65 [cito:cites] [cito:citesAsAuthority] [cito:extends]
14. Clevert D-A, Le T, Winter R, Montanari F (2021) Img2Mol—accurate SMILES recognition from molecular graphical depictions. Chem Sci. https://doi.org/10.1039/D1SC01839F [cito:cites] [cito:agreesWith]
15. Weir H, Thompson K, Woodward A, Choi B, Braun A, Martínez TJ (2021) ChemPix: automated recognition of hand-drawn hydrocarbon structures using deep learning. Chem Sci 12:10622–10633 [cito:cites]
16. Oldenhof M, Arany A, Moreau Y, Simm J (2020) Chemgrapher: optical graph recognition of chemical compounds by deep learning. J Chem Inf Model 60:4506–4517 [cito:cites]
17. Zhang X-C, Yi J-C, Yang G-P, Wu C-K, Hou T-J, Cao D-S (2022) ABC-Net: a divide-and-conquer based deep learning architecture for SMILES recognition from molecular images. Brief Bioinform. https://doi.org/10.1093/bib/bbac033 [cito:cites]
18. Khokhlov I, Krasnov L, Fedorov MV, Sosnin S (2022) Image2SMILES: transformer-based molecular optical recognition engine. Chem Methods. https://doi.org/10.1002/cmtd.202100069 [cito:cites]
19. Osra (2022) https://sourceforge.net/p/osra/wiki/Validation/. Accessed 30 Mar 2022 [cito:cites] [cito:citesAsDataSource]
20. Ouyang TY, Davis R (2007) Recognition of hand drawn chemical diagrams. AAAI 7:846–851 [cito:cites]
21. Ramel J-Y, Boissier G, Emptoz H (1999) Automatic reading of handwritten chemical formulas from a structural representation of the image. In: Proceedings of the 5th International Conference on Document Analysis and Recognition, ICDAR '99 (Cat. No.PR00318), pp 83–86 [cito:cites]
22. Vision Arcanum: InkToMolecule online. https://visionarcanum.com/ink2mol/. Accessed 30 Mar 2022 [cito:cites]
23. Weininger D (1988) SMILES, a chemical language and information system. 1. Introduction to methodology and encoding rules. J Chem Inf Comput Sci 28:31–36 [cito:usesMethodIn]
24. Kim S, Chen J, Cheng T et al (2021) PubChem in 2021: new data content and improved web interfaces. Nucleic Acids Res 49:D1388–D1395 [cito:citesAsDataSource] [cito:usesDataFrom]
25. Ashton M, Barnard J, Casset F, Charlton M, Downs G, Gorse D, Holliday J, Lahana R, Willett P (2002) Identification of diverse database subsets using property-based and fragment-based molecular descriptions. Quant struct-act relatsh 21:598–604 [cito:usesMethodIn] [cito:cites]
26. Morgan HL (1965) The generation of a unique machine description for chemical structures-A technique developed at chemical abstracts service. J Chem Doc 5:107–113 [cito:usesMethodIn] [cito:cites]
27. Mayfield J, Swain M, Willighagen E (2022) CDK Depict. In: GitHub. https://github.com/cdk/depict. Accessed 4 Mar 2022 [cito:cites] [cito:usesMethodIn]
28. Steinbeck C, Han Y, Kuhn S, Horlacher O, Luttmann E, Willighagen E (2003) The chemistry development kit (CDK): an open-source Java library

for chemo- and bioinformatics. J Chem Inf Comput Sci 43:493–500 [cito:usesMethodIn]

29. Jacobsen A, de Miranda AR, Juty N et al (2020) FAIR principles: Interpretations and implementation considerations. Data Intelligence 2:10–29 [cito:agreesWith]

Publisher's Note

Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

2.4 Publication D: Open data and algorithms for open science in AI-driven molecular informatics

Brinkhaus, H.O.[1], Rajan, K.[2], Schaub, J.[3], Zielesny, A.[4], Steinbeck, C.[5]

Curr. Opin. Struct. Biol. 79, 102542 (2023)

DOI: 10.1016/j.sbi.2023.102542

Table 5: Author contributions for Publication D

Author No	**1**	**2**	**3**	**4**	**5**
Conceptual research design	x	x	x	x	x
Planning of research activities	x	x	x	x	x
Information collection	x	x	x		
Manuscript writing	x	x	x	x	x
Suggested publication equivalence value	0.5				

***underlined author numbers refer to involved doctoral students**

Available online at www.sciencedirect.com

ScienceDirect

Current Opinion in
Structural Biology

Open data and algorithms for open science in AI-driven molecular informatics

Henning Otto Brinkhaus[1], Kohulan Rajan[1], Jonas Schaub[1], Achim Zielesny[2] and Christoph Steinbeck[1]

Abstract
Recent years have seen a sharp increase in the development of deep learning and artificial intelligence-based molecular informatics. There has been a growing interest in applying deep learning to several subfields, including the digital transformation of synthetic chemistry, extraction of chemical information from the scientific literature, and AI in natural product-based drug discovery. The application of AI to molecular informatics is still constrained by the fact that most of the data used for training and testing deep learning models are not available as FAIR and open data. As open science practices continue to grow in popularity, initiatives which support FAIR and open data as well as open-source software have emerged. It is becoming increasingly important for researchers in the field of molecular informatics to embrace open science and to submit data and software in open repositories. With the advent of open-source deep learning frameworks and cloud computing platforms, academic researchers are now able to deploy and test their own deep learning models with ease. With the development of new and faster hardware for deep learning and the increasing number of initiatives towards digital research data management infrastructures, as well as a culture promoting open data, open source, and open science, AI-driven molecular informatics will continue to grow. This review examines the current state of open data and open algorithms in molecular informatics, as well as ways in which they could be improved in future.

Addresses
[1] Institute for Inorganic and Analytical Chemistry, Friedrich Schiller University Jena, Lessingstr. 8, 07743 Jena, Germany
[2] Institute for Bioinformatics and Chemoinformatics, Westphalian University of Applied Sciences, August-Schmidt-Ring 10, 45665 Recklinghausen, Germany

Corresponding author: Steinbeck, Christoph (christoph.steinbeck@uni-jena.de)

Current Opinion in Structural Biology 2023, **79:**102542

This review comes from a themed issue on **Artificial Intelligence (AI) Methodology in Structural Biology (2023)**

Edited by **Andreas Bender**, **Chris de Graaf** and **Noel O'Boyle**

For complete overview of the section, please refer the article collection - Artificial Intelligence (AI) Methodology in Structural Biology (2023)

Available online 17 February 2023

https://doi.org/10.1016/j.sbi.2023.102542

Introduction

Considerable improvements in artificial intelligence (AI) research through the introduction of deep neural networks promise to transform society [1–4] and the way research is conducted [5,6]. However, in most areas of molecular informatics, the amount of training data available is insufficient for the use of today's most powerful deep neural network architectures, which demonstrate superior performance only by training with large amounts of data [7]. In addition, a thorough assessment of a model's true predictive performance in practice is a rare exception (e.g. the Critical Assessment of Protein Structure Prediction (CASP) [8]).

Because of this lack of accessible experimental data [9,10], machine learning predictions in chemistry are generally too error-prone to realize the potential of the new methods at this time. This necessitates a change in the way chemists publish their data and the type of data published [11,12]. The call for open data, open source, and open science (ODOSOS) in chemistry is not new [13,14], but with the advent of more powerful data-driven algorithms, it has never been more important.

Journals and funders demanding the deposition of research data and the necessary establishment of suitable research data infrastructures will inevitably alleviate the data shortage problem in the future [15,16]. The German government, for example, has recently decided to implement and long-term-fund a national research data infrastructure (Nationale Forschungsdateninfrastruktur, NFDI) [17] with 30 consortia in all areas of science, collaboratively developing open research data management (RDM) e-infrastructures, coordinated by an umbrella process and a joint directorate. One of those consortia is NFDI4Chem which is building an RDM e-infrastructure for chemistry that follows FAIR data principles [18] to make chemical data findable, accessible, interoperable, and reusable [19,20]. One flagship project of NFDI4Chem is nmrXiv, an open and FAIR repository and analysis platform for NMR spectroscopy data [21].

In recent years, advances in artificial intelligence and data-driven applications in molecular informatics have provided a glimpse into the magnitude of future accomplishments, which have made open data a necessity for machine learning algorithms. Here, we attempt to present some of the major milestones of the past years and discuss obstacles that are yet to be overcome to enable similar AI-driven progress in (nearly) every area of chemistry.

The importance of openly available resources and data

One cause of the dissatisfying data shortage situation has been the lack of a culture of data deposition and sharing in chemistry in the past, where at least from the early 1990s onwards, with the advent of the internet, widespread data deposition and sharing would have been possible. There have been notable exceptions, such as the crystallography community, that have developed data deposition cultures even earlier. Both small molecules and biomacromolecule structures have been and are being deposited in the Protein Data Bank (PDB) [22,23] and the Cambridge Crystallographic Database (CCD) [24]. Of particular note, the open PDB in combination with the openly available protein sequence information (for multiple sequence alignments) formed the basis for the outstanding success of the AlphaFold protein 3D structure prediction system [5]. Similarly, open databases such as PubChem [25], ChEMBL [26], ChEBI [27], Drugbank [28], the Human Metabolome Database (HMDB) [29], the Collection of Open Natural Products (COCONUT) [30], the Natural Products Atlas [31], the Natural Products Magnetic Resonance Database [32], and ZINC [33] fundamentally broaden the research opportunities [34]. The PubChem database is used by millions of users every month [35]. An example for the usage of the referenced databases is the creation of a classifier that determines whether a Natural Product (NP) originates from funghi, plants, or bacteria based on its chemical structure with data obtained from the COCONUT database [36]. The ZINC database has recently been used for the in silico determination of drug candidates that inhibit the main protease of SARS-CoV-2 [37].

Another crucial aspect is the availability of open software libraries to handle and process chemical information, like the Chemistry Development Kit (CDK) [38], Indigo [39], RDKit [40], or OpenBabel [41], as well as the recently published Python-based Informatics Kit for Analysing Chemical Units (PIKAChU) [42]. Without these open-source projects, the research community would lack basic tools for programmatically reading, modifying, and processing chemical information. Accordingly, they are fundamental for every researcher in the field of molecular informatics.

Molecular string representations, such as DeepSMILES [43] and SELFIES [44], enable processing chemical structures using models like transformers that are designed to process sequential data. Recently, a study investigated the performance of transformers on different tasks using SMILES, DeepSMILES, and SELFIES. The amount of returned invalid chemical structures could be decreased when using DeepSMILES and especially SELFIES compared to SMILES, although the overall best performance was achieved using SMILES [45].

Without open libraries such as Tensorflow [46] and Pytorch [47] for the implementation and training of neural networks as well as the ubiquitous availability of Graphical Processing Units (GPU) and Tensor Processing Units (TPU) in cloud environments [48], the big leaps in molecular AI research would not have been possible.

An approach to the protein folding problem - AlphaFold

The problem of protein folding is considered one of the fundamental challenges of molecular biology because a large number of degrees of freedom of bonds and atoms in a protein leads to a combinatorial explosion in the number of possible low-energy arrangements [49]. In 2020, the *DeepMind* team announced a widely recognised breakthrough in the prediction of spatial protein 3D structures from their amino acid sequence with their deep learning-based system *AlphaFold* [5]. The system participated in the 13th and 14th Critical Assessment of Protein Structure Prediction (CASP) competition [8], outperforming all competitors. Since then, it has been made openly available and used to fill the open *AlphaFold Protein Structure Database* [50] which contains more than 200 million predicted protein 3D structures, covering nearly every known protein on earth [51]. Within a short period of time, the structures of 98.5% of the human proteome have been predicted using *AlphaFold*, while the previous decades of experimental research yielded 17% [52]. The system was trained on structural data openly deposited in the Protein Data Bank [22,23], which was founded and announced in 1971 [53]. The success story of *AlphaFold* illustrates what is possible today when researchers are able to access the data that scientists have produced over the course of 50 years.

It is important to mention that challenges like the prediction of the relative positions of protein domains and their changes when an external stimulus is applied remain partially unsolved. Additionally, the transition from disordered to ordered domain states cannot be elucidated using *AlphaFold*, and it is limited to structures with less than 2700 amino acids [54]. Nevertheless, the high impact of its accurate protein structure predictions is indisputable [55]. For example, the predicted structural information about nucleoporins has been combined with cryo-electron tomography (cryo-ET) to generate a model that precisely explains 90% of the scaffold of the human nuclear pore complex (NPC) [56]. Another example is the identification of tens of thousands of unknown potential binding sites for iron-sulfur clusters and zinc ions in more than 360,000 proteins [57].

Digital transformation of synthetic chemistry

Similar to other fields, the foundation for successful machine learning applications in synthetic organic chemistry is the availability of extensive experimental data [58]. Recently, Strieth-Kalthoff et al. demonstrated the benefit that emerges from the usage of real experimental data for machine learning-based chemical yield predictions [12] while the prediction of reaction outcomes and yields remains a challenge in general [59]. Nonetheless, there have been impressive developments using attention-based deep learning methods to explore the chemical reaction space [60]. Schwaller et al. trained a transformer to predict chemical reaction outcomes with state-of-the-art results [61]. The resulting model which is referred to as *molecular transformer* was then used in combination with hypergraph exploration to automatically plan retrosynthesis routes [62]. Since then, the *molecular transformer* has been extended to predict the products of enzymatic reactions [63]. Based on the aforementioned retrosynthesis planning system, Probst et al. have published a biocatalysed synthesis planning system [64].

Schwaller et al. have also shown that the attention matrix weights of transformers that have been trained on unlabelled chemical reaction data can be used to determine accurate atom mappings [65]. Additionally, they demonstrated that attention-based models are highly suitable for the classification of chemical reactions [66]. Similar model architectures were successfully used to generate specific synthesis instructions [67] and to determine the yield of a given chemical reaction formula [68]. Andronov et al. successfully demonstrated the prediction of reagents based on given reaction SMILES strings using transformers. They were then able to use the reagent prediction model to fill in missing reagents in incomplete reaction data from US patents leading to an improved state-of-the-art model [61] for the prediction of reaction products [69]. Recently, Rohrbach et al. demonstrated the translation of synthesis protocols in the literature into a standardized chemical language, which could then be executed by their automated synthesis system [70].

Again, the described advances are exemplary cases of the synergy of deep learning-based models and the availability of training data. There are datasets extracted from US patents [66,71–74], the scientific literature [75], and high-throughput experiments (HTE) [76] available [60]. Recently, the Open Reaction Database (ORD) has been launched as a platform to replace unstructured reaction data in the supporting information of publications [77]. If it is accepted by the research community, the ORD may become a part of the solution to problems caused by the aforementioned lack of data and report bias [11,12]. Providing structured data in standardized formats may become a key step towards the digital transformation of synthetic chemistry.

Extraction of chemical information from the scientific literature

Besides enforcing FAIR data publication standards today and in the near future, it is important to tackle the damage that has already been done by publishing chemical data almost exclusively in a human-readable form with unstructured text and images in the past decades. The advances in the fields of natural language processing (NLP) [78–80] and computer vision (CV) [81–83] have made a new generation of chemical literature mining tools possible. These can be considered AI-driven solutions that enable further AI-driven advances by making concealed data accessible in structured, machine-readable formats.

The field of optical chemical structure recognition (OCSR) deals with the translation of images of chemical structures as they are published in the scientific literature into machine-readable representations of the underlying molecular graph [84,85]. In the past two years, a variety of deep learning-based OCSR methods [86–89] has been published, where *DECIMER Image-Transformer* [90], *Img2Mol* [91] and *SwinOCSR* [92] provide openly available source code and trained models. For the segmentation of chemical structure images from whole pages, the open-source tool *DECIMER Segmentation* can be used [93]. With the publication of the open-source depiction generation tool RanDepict, efforts have been made to standardize and diversify the training data for deep learning-based OCSR tools [94]. The newest version of DECIMER was trained on more than 400 Million images using the latest Tensor Processing Units [95] available on the Google cloud platform. Currently, DECIMER performs with an accuracy rate of above 90% and is regarded as an important point of reference for future work [85]. Without open databases like PubChem, where one can download over 100 million chemical structures for free, this would not have been possible.

Since its original release in 2016, the chemical literature mining toolkit *ChemDataExtractor* [96] has been continuously developed [97,98]. The highly adaptable toolkit uses user-defined models of the information to be extracted in a pipeline with readers for different publisher formats and a system for interdependency resolution with a set of parsers and a sophisticated chemical named entity recognition system [99] to extract chemical information in a structured data format [97]. In the past years, ChemDataExtractor has been extensively used to automatically generate databases about refraction indices and dielectric constants [100], battery material properties [101], properties of semiconductors for building solar cells [102], magnetic properties [103], as well as UV/Vis spectra [104].

In addition to the technical obstacles, scientific publishers hinder literature mining essentially by hiding publications behind paywalls and limiting the number of publications that can be downloaded and used even if a subscription is available. Some publishers like Elsevier offer markup versions of their publications for text mining purposes to academic researchers [105], but there is a long way to go to truly make all published chemical information available. In 2018, an international group of research funders announced the initiative *Plan S* which requires scientists who benefit from their funding to publish in open-access journals [106]. Recently, the US government announced that they will require all publicly funded research to be openly accessible from 2026 on [107]. With RDM e-infrastructures being established as the mandatory scientific data publication standard, the kind of literature mining methods described herein will become obsolete in the future. For now, they are indispensable for artificially intelligent data-driven applications.

AI in natural product-based drug discovery

The field of drug discovery has shifted towards implementing approaches based on the analysis of large amounts of data and deep learning [108]. As a result of the growing demand for efficient new drugs, the field has experienced rapid growth in the last few years. NP are attractive to drug developers due to their availability and their potential affinity to protein drug targets [109,110].

There have been significant advances in various areas of the field, such as the prediction of biochemical effects of NP based on their molecular structure [111], in the field of genome mining for the discovery of bioactive compounds [112], the mining of mass spectrometry-based metabolomics data [113], and integrative approaches that combine metabolomics and genomics data [114].

The initial hope that large-scale data analysis in the different omics-related research fields would boost the drug discovery rate has not yet materialised [115], but the methods are progressing continuously. The open access to databases and repositories such as MetaboLights [116], the HMDB [29], the Metabolomics Workbench [117], and METASPACE [118] is crucial for the identification of metabolites and NP [112]. In 2021, the Paired Omics Data Platform (PODP) was launched as a community-driven platform that provides linked metabolome and genome data according to the FAIR principles [119].

NP-based drug discovery has greatly benefited from models developed for NLP [120]. For example, in 2021, Huang et al. published *MolTrans*, a state-of-the-art deep learning-based framework for the in silico prediction of Drug—Protein Interactions (DPI) [121]. In the following year, Wang et al. presented their structure-aware multimodal deep DPI prediction model *STAMP-DPI*, which outperforms *MolTrans*. The tool has been published along a large high-quality training and benchmarking dataset [122]. The adaptation of sequence models like the transformer [78] for AI-based drug design requires large amounts of well-curated, high-quality data.

The recent development in the field of deep generative models helps researchers generate molecules with desired properties [123], but a model that can generalise well and can generate molecules with desirable properties requires a large amount of training data. When dealing with artificially generated structures, it is also necessary to consider their synthetic accessibility. To successfully use deep learning on published NP structures, well-curated data is essential. Published data resources are often incomplete, inaccessible, or no longer available [124] which makes available resources like the Natural Products Atlas [31], LOTUS [125], and COCONUT [30] even more important.

The development of deep learning-based models has assisted the advancement of drug discovery overall, with more advancements being made in the development of models and increasing access to open data and open databases helping this field grow. We hope that the research community will continue to actively contribute to openly available data sources to enable further progress in the field.

Conclusions

The developments of the past years demonstrate the potential of data-driven machine learning applications in the field of molecular informatics in an impressive manner [5,65,70]. An obvious requirement to benefit from this development is the availability of open structured experimental data [11,12]. The integration of open data infrastructures will enable AI to be used in nearly every field of chemistry. The application of deep learning methodologies and the sharing of code and data in the field of chemistry are still in their early stages and require more community standards to be developed. Many of the models are still being trained from scratch using in-house servers and GPUs, which is a time-consuming and restrictive process. The rapid growth of the field will be enabled by the sharing of already-trained models and curated data with the public. When sharing code or data, high quality and data standards must be maintained [126]. Using the public cloud infrastructures will readily allow researchers to take advantage of the latest developments in hardware and software, which will lead to faster growth and a reduction in energy consumption. There are several initiatives working continuously to implement open data, open-source, and open science in their individual research area [13,14,17,18,20,21,77,106,107,127,128]. Fueled by the availability of more and more open research data, AI-powered molecular informatics will be a key driver of the digital transformation of chemistry in the coming years.

Author contributions

HOB, KR and JS conducted the literature review and wrote the article. CS and AZ conceived the study and supervised the work. All authors read and approved the final manuscript.

Declaration of competing interest

AZ is co-founder of GNWI—Gesellschaft für naturwissenschaftliche Informatik mbH, Dortmund, Germany.

Data availability

No data was used for the research described in the article.

Acknowledgements

HOB, JS and CS acknowledge funding by the Carl-Zeiss-Foundation. KR acknowledges the funding by the German Research Foundation within the framework CRC1127 ChemBioSys.

List of abbreviations

AI	Artificial Intelligence
CASP	Critical Assessment of protein Structure Prediction
CCD	Cambridge Crystallographic Database
CDK	Chemistry Development Kit
cryo-ET	cryo-Electron Tomography
COCONUT	COlleCtion of Open Natural Products
CV	Computer Vision
DECIMER	Deep lEarning for Chemical ImagE Recognition
DPI	Drug−Protein Interaction
FAIR	Findable, Accessible, Interoperable, and Reusable
GPU	Graphics Processing Unit
HTE	High-Throughput Experiments
HMDB	Human Metabolome DataBase
NFDI	Nationale ForschungsDatenInfrastruktur (National Research Data Infrastructure)
NFDI4Chem	National Research Data Infrastructure for Chemistry
NLP	Natural Language Processing
NP	Natural Products
NP-MRD	Natural Products Magnetic Resonance Database
NPC	Nuclear Pore Complex
OCSR	Optical Chemical Structure Recognition
ODOSOS	Open Data, Open Source, and Open Science
ORD	Open Reaction Database
PDB	Protein DataBank
PODP	Paired Omics Data Platform
PIKAChU	Python-based Informatics Kit for Analysing CHemical Units
RDM	Research Data Management
TPU	Tensor Processing Unit

References

Papers of particular interest, published within the period of review, have been highlighted as:

* of special interest

1. Silver D, Schrittwieser J, Simonyan K, Antonoglou I, Huang A, Guez A, Hubert T, Baker L, Lai M, Bolton A, *et al.*: **Mastering the game of Go without human knowledge**. *Nature* 2017, **550**: 354−359.

2. Gupta A, Anpalagan A, Guan L, Khwaja AS: **Deep learning for object detection and scene perception in self-driving cars: survey, challenges, and open issues**. *Array* 2021, **10**:100057.

3. Ramesh A, Dhariwal P, Nichol A, Chu C, Chen M: **Hierarchical text-conditional image generation with CLIP latents**. *arXiv [csCV]* 2022.

4. Rombach Robin, Blattmann Andreas, Lorenz Dominik, Esser Patrick, Ommer B: **High-resolution image synthesis with latent diffusion models**. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*; 2022: 10684−10695.

5. * Jumper J, Evans R, Pritzel A, Green T, Figurnov M, Ronneberger O, Tunyasuvunakool K, Bates R, Žídek A, Potapenko A, *et al.*: **Highly accurate protein structure prediction with AlphaFold**. *Nature* 2021, **596**:583−589.

The first ever state-of-the-art neural network model capable of predicting the structure of a protein with great accuracy. The authors of this paper demonstrate that the model is capable of predicting the structure of a protein even when there are no similar structures available.

6. Kirkpatrick J, McMorrow B, Turban DHP, Gaunt AL, Spencer JS, Matthews AGDG, Obika A, Thiry L, Fortunato M, Pfau D, *et al.*: **Pushing the frontiers of density functionals by solving the fractional electron problem**. *Science* 2021, **374**:1385−1389.

7. Chuang KV, Gunsalus LM, Keiser MJ: **Learning molecular representations for medicinal chemistry**. *J Med Chem* 2020, **63**:8705−8722.

8. Kryshtafovych A, Schwede T, Topf M, Fidelis K, Moult J: **Critical assessment of methods of protein structure prediction (CASP)-Round XIV**. *Proteins* 2021, **89**:1607−1617.

In this paper, the authors mention that the deep-learning-based method has a higher accuracy for predicting protein structures compared to all previous approaches and it is on par with experimental results in terms of accuracy.

9. Bajorath J: **State-of-the-art of artificial intelligence in medicinal chemistry**. *Future Sci OA* 2021, **7**. FSO702.

10. Tripathi MK, Nath A, Singh TP, Ethayathulla AS, Kaur P: **Evolving scenario of big data and Artificial Intelligence (AI) in drug discovery**. *Mol Divers* 2021, **25**:1439–1460.

11. Cole JM: **The chemistry of errors**. *Nat Chem* 2022, **14**:973–975.
The author of this paper states that in order to improve predictions with machine-learning methods, it is necessary to have access to a substantial amount of experimental data.

12. Strieth-Kalthoff F, Sandfort F, Kühnemund M, Schäfer FR, Kuchen H, Glorius F: **Machine learning for chemical reactivity: the importance of failed experiments**. *Angew Chem Int Ed Engl* 2022, **61**:e202204647.

13. Guha R, Howard MT, Hutchison GR, Murray-Rust P, Rzepa H, Steinbeck C, Wegner J, Willighagen EL: **The Blue Obelisk-interoperability in chemical informatics**. *J Chem Inf Model* 2006, **46**:991–998.

14. O'Boyle NM, Guha R, Willighagen EL, Adams SE, Alvarsson J, Bradley J-C, Filippov IV, Hanson RM, Hanwell MD, Hutchison GR, *et al.*: **Open data, open source and open standards in chemistry: the blue obelisk five years on**. *J Cheminf* 2011, **3**:37.

15. * Schymanski EL: **Bolton EE: FAIR chemical structures in the journal of cheminformatics**. *J Cheminf* 2021, **13**:50.
The purpose of this paper is to emphasize the importance of openly sharing chemical information and structural data in the field of cheminformatics. Providing data according to FAIR principles enhances the journal's commitment to open science, and both readers and authors will benefit from such an initiative.

16. Zdrazil B, Guha R: **Diversifying cheminformatics**. *J Cheminf* 2022, **14**:25.

17. Hartl N, Wössner E, Sure-Vetter Y: **Nationale Forschungsdateninfrastruktur (NFDI)**. *Informatik-Spektrum* 2021, **44**:370–373.

18. Steinbeck C, Koepler O, Bach F, Herres-Pawlis S, Jung N, Liermann J, Neumann S, Razum M, Baldauf C, Biedermann F, *et al.*: **NFDI4Chem-Towards a national research data infrastructure for chemistry in Germany**. *Research Ideas and Outcomes* 2020, **6**:e55852.

19. Rzepa HS: **The long and winding road towards FAIR data as an integral component of the computational modelling and dissemination of chemistry**. *Isr J Chem* 2022, **62**:e202100034.

20. Herres-Pawlis S, Liermann JC, Koepler O: **Research data in chemistry – results of the first NFDI4Chem community survey**. *Z Anorg Allg Chem* 2020, **646**:1748–1757.
With the growing demand for open and FAIR data, research data management is becoming increasingly important in chemistry. NFDI4Chem is a first-of-its-kind national research data infrastructure initiative aimed at providing a public research data management platform for researchers to store and share the data they produce in the field of chemistry.

21. NFDI4Chem: **nmrXiv - open, FAIR and Consensus-Driven NMR spectroscopy data repository and analysis platform**. In *nmrXiv - Open, FAIR and Consensus-Driven NMR spectroscopy data repository and analysis platform*; 2022.
An ongoing important project of NFDI4Chem is nmrXiv, an open and FAIR repository and analysis platform for NMR spectroscopy data. This is the first open platform of its kind made available to the public.

22. wwPDB consortium: **Protein Data Bank: the single global archive for 3D macromolecular structure data**. *Nucleic Acids Res* 2019, **47**:D520–D528.

23. Burley SK, Bhikadiya C, Bi C, Bittrich S: **RCSB Protein Data Bank: powerful new tools for exploring 3D structures of biological macromolecules for basic and applied research and education in fundamental**. *Nucleic acids* 2021.

24. Groom CR, Bruno IJ, Lightfoot MP, Ward SC: **The Cambridge structural database**. *Acta Crystallogr B Struct Sci Cryst Eng Mater* 2016, **72**:171–179.

25. Kim S, Chen J, Cheng T, Gindulyte A, He J, He S, Li Q, Shoemaker BA, Thiessen PA, Yu B, *et al.*: **PubChem in 2021: new data content and improved web interfaces**. *Nucleic Acids Res* 2021, **49**:D1388–D1395.

26. Gaulton A, Hersey A, Nowotka M, Bento AP, Chambers J, Mendez D, Mutowo P, Atkinson F, Bellis LJ, Cibrián-Uhalte E, *et al.*: **The ChEMBL database in 2017**. *Nucleic Acids Res* 2017, **45**:D945–D954.

27. Hastings J, Owen G, Dekker A, Ennis M, Kale N, Muthukrishnan V, Turner S, Swainston N, Mendes P, Steinbeck C: **ChEBI in 2016: improved services and an expanding collection of metabolites**. *Nucleic Acids Res* 2016, **44**:D1214–D1219.

28. Wishart DS, Feunang YD, Guo AC, Lo EJ, Marcu A, Grant JR, Sajed T, Johnson D, Li C, Sayeeda Z, *et al.*: **DrugBank 5.0: a major update to the DrugBank database for 2018**. *Nucleic Acids Res* 2018, **46**:D1074–D1082.

29. Wishart DS, Guo A, Oler E, Wang F, Anjum A, Peters H, Dizon R, Sayeeda Z, Tian S, Lee BL, *et al.*: **HMDB 5.0: the human metabolome database for 2022**. *Nucleic Acids Res* 2022, **50**: D622–D631.

30. * Sorokina M, Merseburger P, Rajan K, Yirik MA: **Steinbeck C: COCONUT online: collection of open natural products database**. *J Cheminf* 2021, **13**:2.
The COlleCtion of Open Natural prodUcTs (COCONUT) is currently the largest open database available for natural products. The database currently contains data from 53 natural products databases, and it is one of the most active databases in the field.

31. van Santen JA, Poynton EF, Iskakova D, McMann E, Alsup TA, Clark TN, Fergusson CH, Fewer DP, Hughes AH, McCadden CA, *et al.*: **The Natural Products Atlas 2.0: a database of microbially-derived natural products**. *Nucleic Acids Res* 2022, **50**:D1317–D1323.

32. Wishart DS, Sayeeda Z, Budinski Z, Guo A, Lee BL, Berjanskii M, Rout M, Peters H, Dizon R, Mah R, *et al.*: **NP-MRD: the natural products magnetic resonance database**. *Nucleic Acids Res* 2022, **50**:D665–D677.

33. Irwin JJ, Tang KG, Young J, Dandarchuluun C, Wong BR, Khurelbaatar M, Moroz YS, Mayfield J, Sayle RA: **ZINC20-A free ultralarge-scale chemical database for ligand discovery**. *J Chem Inf Model* 2020, **60**:6065–6073.

34. Wegner JK, Sterling A, Guha R, Bender A, Faulon J-L, Hastings J, O'Boyle N, Overington J, Van Vlijmen H, Willighagen E: *Cheminformatics. Commun ACM* 2012, **55**:65–75.

35. Kim S, Cheng T, He S, Thiessen PA, Li Q, Gindulyte A, Bolton EE: **PubChem protein, gene, pathway, and taxonomy data collections: bridging biology and chemistry through target-centric views of PubChem data**. *J Mol Biol* 2022, **434**: 167514.

36. Capecchi A, Reymond J-L: **Classifying natural products from plants, fungi or bacteria using the COCONUT database and machine learning**. *J Cheminf* 2021, **13**:82.

37. Mathpal S, Joshi T, Sharma P, Joshi T, Pundir H, Pande V, Chandra S: **A dynamic simulation study of FDA drug from zinc database against COVID-19 main protease receptor**. *J Biomol Struct Dyn* 2022, **40**:1084–1100.

38. Willighagen EL, Mayfield JW, Alvarsson J, Berg A, Carlsson L, Jeliazkova N, Kuhn S, Pluskal T, Rojas-Chertó M, Spjuth O, *et al.*: **The Chemistry Development Kit (CDK) v2.0: atom typing, depiction, molecular formulas, and substructure searching**. *J Cheminf* 2017:9.

39. Pavlov D, Rybalkin M, Karulin B, Kozhevnikov M, Savelyev A, Churinov A: **Indigo: universal cheminformatics API**. *J Cheminf* 2011, **3**. P4.

40. Landrum G, Tosco P, Kelleyet B, *et al.*: ***rdkit: 2022_03_3 (Q1 2022) Release.*** 2022, https://doi.org/10.5281/zenodo.7541264.

41. O'Boyle NM, Banck M, James CA, Morley C, Vandermeersch T, Hutchison GR: **Open Babel: an open chemical toolbox**. *J Cheminf* 2011, **3**:33.

42. Terlouw BR, Vromans SPJM, Medema MH: **PIKAChU: a Python-based informatics kit for analysing chemical units**. *J Cheminf* 2022, **14**:34.

The PIKAChU cheminformatics library is a pure python library that can be used for many cheminformatics analyses using python. It is the first-ever cheminformatics library to be written entirely in Python.

43. O'Boyle N, Dalke A: *DeepSMILES: An Adaptation of SMILES for Use in Machine-Learning of Chemical Structures*. 2018, https://doi.org/10.26434/chemrxiv.7097960.v1.

44. * Krenn M, Ai Q, Barthel S, Carson N, Frei A, Frey NC, Friederich P, Gaudin T, Gayle AA, Jablonka KM, *et al.*: **SELFIES and the future of molecular string representations**. *Patterns Prejudice* 2022, **3**:100588.

String representations are widely used in deep learning models in chemistry in order to learn and interpret chemical structures. The most widely used string representation, SMILES, is not designed for deep learning tasks. To address this issue, the authors of this paper have developed a new machine-readable string representation, SELF-referencIng Embedded Strings (SELFIES). Moreover, the paper examines the different string representations that are currently available, their shortcomings, and the potential for future development.

45. Rajan K, Steinbeck C, Zielesny A: **Performance of chemical structure string representations for chemical image recognition using transformers**. *Digital Discovery* 2022, **1**:84–90.

46. Abadi M, Agarwal A, Barham P, Brevdo E, Chen Z, Citro C, Corrado GS, Davis A, Dean J, Devin M, *et al.*: **TensorFlow: large-scale machine learning on heterogeneous distributed systems**. *arXiv [csDC]* 2016.

47. Paszke A, Gross S, Massa F, Lerer A, Bradbury J, Chanan G, Killeen T, Lin Z, Gimelshein N, Antiga L, *et al.*: **PyTorch: an imperative style, high-performance deep learning library**. *arXiv [csLG]* 2019.

48. You Y, Zhang Z, Hsieh C, Demmel J, Keutzer K: **Fast deep neural network training on distributed systems and cloud TPUs**. *IEEE Trans Parallel Distr Syst* 2019, **30**:2449–2462.

49. Levinthal C: *How to fold graciously. Mossbauer spectroscopy in biological systems proceedings*. University of Illinois Bulletin; 1969.

50. Varadi M, Anyango S, Deshpande M, Nair S, Natassia C, Yordanova G, Yuan D, Stroe O, Wood G, Laydon A, *et al.*: **AlphaFold Protein Structure Database: massively expanding the structural coverage of protein-sequence space with high-accuracy models**. *Nucleic Acids Res* 2022, **50**:D439–D444.

51. * Callaway E: **"The entire protein universe": AI predicts shape of nearly every known protein**. *Nature* 2022, **608**:15–16.

Thus far, AlphaFold, a deep-learning algorithm, has predicted nearly 200 million protein structures for over one million species, meaning that nearly every protein on the planet has a structure that has been predicted by AlphaFold. This article highlights the significance of this contribution to the natural sciences.

52. Tunyasuvunakool K, Adler J, Wu Z, Green T, Zielinski M, Žídek A, Bridgland A, Cowie A, Meyer C, Laydon A, *et al.*: **Highly accurate protein structure prediction for the human proteome**. *Nature* 2021, **596**:590–596.

53. **Data, Crystallography: protein data bank**. *Nat New Biol* 1971.

54. David A, Islam S, Tankhilevich E, Sternberg MJE: **The AlphaFold database of protein structures: a biologist's guide**. *J Mol Biol* 2022, **434**:167336.

55. Varadi M, Velankar S: **The impact of AlphaFold Protein Structure Database on the fields of life sciences**. *Proteomics* 2022.

56. Mosalaganti S, Obarska-Kosinska A, Siggel M, Taniguchi R, Turoňová B, Zimmerli CE, Buczak K, Schmidt FH, Margiotta E, Mackmull M-T, *et al.*: **AI-based structure prediction empowers integrative structural analysis of human nuclear pores**. *Science* 2022, **376**. eabm9506.

57. Wehrspan ZJ, McDonnell RT, Elcock AH: **Identification of iron-sulfur (Fe-S) cluster and zinc (Zn) binding sites within proteomes predicted by DeepMind's AlphaFold2 program dramatically expands the metalloproteome**. *J Mol Biol* 2022, **434**:167377.

58. Segler MHS, Preuss M, Waller MP: **Planning chemical syntheses with deep neural networks and symbolic AI**. *Nature* 2018, **555**:604–610.

59. Davies IW: **The digitization of organic synthesis**. *Nature* 2019, **570**:175–181.

60. Schwaller P, Vaucher AC, Laplaza R, Bunne C, Krause A, Corminboeuf C, Laino T: **Machine intelligence for chemical reaction space**. *Wiley Interdiscip Rev Comput Mol Sci* 2022, https://doi.org/10.1002/wcms.1604.

61. Schwaller P, Laino T, Gaudin T, Bolgar P, Hunter CA, Bekas C, Lee AA: **Molecular transformer: a model for uncertainty-calibrated chemical reaction prediction**. *ACS Cent Sci* 2019, **5**: 1572–1583.

This is a first-of-its-kind data-driven deep-learning model for the prediction of chemical reactions. Using the model presented in this publication, the authors were able to achieve an accuracy of over 98% in classification. In addition, they explain that the learned representations could be used as fingerprints of reactions.

62. Schwaller P, Petraglia R, Zullo V, Nair VH, Haeuselmann RA, Pisoni R, Bekas C, Iuliano A, Laino T: **Predicting retrosynthetic pathways using transformer-based models and a hyper-graph exploration strategy**. *Chem Sci* 2020, **11**:3316–3325.

63. Kreutter D, Schwaller P, Reymond J-L: **Predicting enzymatic reactions with a molecular transformer**. *Chem Sci* 2021, **12**: 8648–8659.

64. Probst D, Manica M, Nana Teukam YG, Castrogiovanni A, Paratore F, Laino T: **Biocatalysed synthesis planning using data-driven learning**. *Nat Commun* 2022, **13**:964.

65. Schwaller P, Hoover B, Reymond J-L, Strobelt H, Laino T: **Extraction of organic chemistry grammar from unsupervised learning of chemical reactions**. *Sci Adv* 2021:7.

66. Schwaller P, Probst D, Vaucher AC, Nair VH, Kreutter D, Laino T, Reymond J-L: **Mapping the space of chemical reactions using attention-based neural networks**. *Nat Mach Intell* 2021, **3**: 144–152.

67. Vaucher AC, Zipoli F, Geluykens J, Nair VH, Schwaller P, Laino T: **Automated extraction of chemical synthesis actions from experimental procedures**. *Nat Commun* 2020, **11**:3601.

68. Schwaller P, Vaucher AC, Laino T, Reymond J-L: **Prediction of chemical reaction yields using deep learning**. *Mach Learn: Sci Technol* 2021, **2**, 015016.

69. Andronov M, Voinarovska V, Andronova N, Wand M, Clevert D-A, Schmidhuber J: **Reagent prediction with a molecular transformer improves reaction data quality**. *ChemRxiv* 2022, https://doi.org/10.26434/chemrxiv-2022-sn2kr.

70. Rohrbach S, Šiaučiulis M, Chisholm G, Pirvan P-A, Saleeb M, Mehr SHM, Trushina E, Leonov AI, Keenan G, Khan A, *et al.*: **Digitization and validation of a chemical synthesis literature database in the ChemPU**. *Science* 2022, **377**:172–180.

71. Coley CW, Barzilay R, Jaakkola TS, Green WH, Jensen KF: **Prediction of organic reaction outcomes using machine learning**. *ACS Cent Sci* 2017, **3**:434–443.

72. Jin W, Coley C, Barzilay R, Jaakkola T: **Predicting organic reaction outcomes with Weisfeiler-Lehman network**. *Adv Neural Inf Process Syst* 2017, **30**.

73. Schwaller P, Gaudin T, Lányi D, Bekas C, Laino T: **"Found in Translation": predicting outcomes of complex organic chemistry reactions using neural sequence-to-sequence models**. *Chem Sci* 2018, **9**:6091–6098.

74. Dai H, Li C, Coley C: **Dai: retrosynthesis prediction with conditional graph logic network**. *Adv Neural Inf Process Syst* 2019, **32,** b.

75. Jiang S, Zhang Z, Zhao H, Li J, Yang Y, Lu B-L, Xia N: **When SMILES smiles, practicality judgment and yield prediction of chemical reaction via deep chemical Language Processing**. *IEEE Access* 2021, **9**:85071–85083.

76. Nielsen MK, Ahneman DT, Riera O, Doyle AG: **Deoxyfluorination with sulfonyl fluorides: navigating reaction space with machine learning**. *J Am Chem Soc* 2018, **140**:5004–5008.

77. Kearnes SM, Maser MR, Wleklinski M, Kast A, Doyle AG, Dreher SD, Hawkins JM, Jensen KF, Coley CW: **The open reaction database**. *J Am Chem Soc* 2021, **143**:18820–18826.

78. Vaswani A, Shazeer N, Parmar N, Uszkoreit J, Jones L, Gomez AN, Kaiser L, Polosukhin I: **Attention is all you need**. *Adv Neural Inf Process Syst* 2017, **30**.

79. Devlin J, Chang M-W, Lee K, Toutanova K: **BERT: pre-training of deep bidirectional transformers for language understanding**. *arXiv [csCL]* 2018.

80. Brown T, Mann B, Ryder N, *et al.*: **Language models are few-shot learners**. *Adv Neural Inf Process Syst* 2020, **33**: 1877–1901.

81. Tan M, Le Q: **EfficientNetV2: smaller models and faster training**. In *Proceedings of the 38th international conference on machine learning*. Edited by Meila M, Zhang T; 18–24 Jul 2021: 10096–10106. PMLR.

82. Liu Z, Lin Y, Cao Y, Hu H, Wei Y, Zhang Z, Lin S, Guo B: **Swin Transformer: Hierarchical Vision Transformer using Shifted Windows**. *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)* 2021:10012–10022.

83. Dosovitskiy A, Beyer L, Kolesnikov A, Weissenborn D, Zhai X, Unterthiner T, Dehghani M, Minderer M, Heigold G, Gelly S, *et al.*: **An image is worth 16x16 words: transformers for image recognition at scale**. *arXiv [csCV]* 2020.

84. * Rajan K, Brinkhaus HO, Zielesny A: **Steinbeck C: a review of optical chemical structure recognition tools**. *J Cheminf* 2020, **12**:60.
This paper summarizes the currently available rule-based and deep learning-based tools for optical chemical structure recognition and explains the need for more open-source and deep learning-based tools in this field. This is the first review published in the field of OCSR.

85. Musazade F, Jamalova N, Hasanov J: **Review of techniques and models used in optical chemical structure recognition in images and scanned documents**. *J Cheminf* 2022, **14**:61.

86. Oldenhof M, Arany A, Moreau Y, Simm J: **ChemGrapher: optical graph recognition of chemical compounds by deep learning**. *J Chem Inf Model* 2020, **60**:4506–4517.

87. Weir H, Thompson K, Woodward A, Choi B, Braun A, Martínez TJ: **ChemPix: automated recognition of hand-drawn hydrocarbon structures using deep learning**. *Chem Sci* 2021, **12**:10622–10633.

88. Yoo S, Kwon O, Lee H: **Image-to-Graph transformers for chemical structure recognition**. In *ICASSP 2022-2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*; 2022:3393–3397.

89. Zhang X-C, Yi J-C, Yang G-P, Wu C-K, Hou T-J, Cao D-S: **ABC-Net: a divide-and-conquer based deep learning architecture for SMILES recognition from molecular images**. *Briefings Bioinf* 2022:23.

90. Rajan K, Zielesny A, Steinbeck C: **DECIMER 1.0: deep learning for chemical image recognition using transformers**. *J Cheminf* 2021, **13**:61.

91. Clevert D-A, Le T, Winter R, Montanari F: **Img2Mol - accurate SMILES recognition from molecular graphical depictions**. *Chem Sci* 2021, https://doi.org/10.1039/D1SC01839F.

92. Xu Z, Li J, Yang Z, Li S, Li H: **SwinOCSR: end-to-end optical chemical structure recognition using a Swin Transformer**. *J Cheminf* 2022, **14**:41.

93. * Rajan K, Brinkhaus HO, Sorokina M, Zielesny A: **Steinbeck C: DECIMER-Segmentation: automated extraction of chemical structure depictions from scientific literature**. *J Cheminf* 2021, **13**:20.
DECIMER -Segmentation is the first and only open deep learning-based algorithm available for the segmentation of chemical structures in the literature. This tool is capable of detecting and segmenting chemical structures from literature with an accuracy of more than 90%. In this field, this is the first paper to explain such an algorithm.

94. Brinkhaus HO, Rajan K, Zielesny A, Steinbeck C: **RanDepict: random chemical structure depiction generator**. *J Cheminf* 2022, **14**:31.

95. Norrie T, Patil N, Yoon DH, Kurian G, Li S, Laudon J, Young C, Jouppi N, Patterson D: **The design process for google's training chips: TPUv2 and TPUv3**. *IEEE Micro* 2021, **41**:56–63.

96. Swain MC, Cole JM: **ChemDataExtractor: a toolkit for automated extraction of chemical information from the scientific literature**. *J Chem Inf Model* 2016, **56**:1894–1904.

97. Mavračić J, Court CJ, Isazawa T, Elliott SR, Cole JM: **ChemDataExtractor 2.0: autopopulated ontologies for materials science**. *J Chem Inf Model* 2021, **61**:4280–4289.

98. Zhu M, Cole JM: **PDFDataExtractor: a tool for reading scientific text and interpreting metadata from the typeset literature in the portable document format**. *J Chem Inf Model* 2022, **62**: 1633–1643.

99. Isazawa T, Cole JM: **Single model for organic and inorganic chemical named entity recognition in ChemDataExtractor**. *J Chem Inf Model* 2022, **62**:1207–1213.

100. Zhao J, Cole JM: **A database of refractive indices and dielectric constants auto-generated using ChemDataExtractor**. *Sci Data* 2022, **9**:192.

101. Huang S, Cole JM: **A database of battery materials auto-generated using ChemDataExtractor**. *Sci Data* 2020, **7**:260.

102. Beard EJ, Cole JM: **Perovskite- and dye-sensitized solar-cell device databases auto-generated using ChemDataExtractor**. *Sci Data* 2022, **9**:329.

103. Court CJ, Cole JM: **Auto-generated materials database of Curie and Néel temperatures via semi-supervised relationship extraction**. *Sci Data* 2018, **5**:180111.

104. Beard EJ, Sivaraman G: **Vázquez-Mayagoitia Á, Vishwanath V, Cole JM: comparative dataset of experimental and computational attributes of UV/vis absorption spectra**. *Sci Data* 2019, **6**:307.

105. Van Noorden R: **Elsevier opens its papers to text-mining**. *Nature* 2014, **506**:17.

106. Else H: **A guide to Plan S: the open-access initiative shaking up science publishing**. *Nature* 2021, https://doi.org/10.1038/d41586-021-00883-6.
This article describes a first-of-its-kind global initiative of scientific funders to require all research conducted under their funding to be published openly. An initiative like this of key stakeholders in the scientific system holds great promise to improve the open availability of research data and findings in the near future.

107. Tollefson J, Van Noorden R: **US government reveals big changes to open-access policy**. *Nature* 2022, **609**:234–235.

108. Jiménez-Luna J, Grisoni F, Weskamp N, Schneider G: **Artificial intelligence in drug discovery: recent advances and future perspectives**. *Expet Opin Drug Discov* 2021, **16**:949–959.

109. Atanasov AG, Zotchev SB, Dirsch VM: **International natural product sciences taskforce, supuran CT: natural products in drug discovery: advances and opportunities**. *Nat Rev Drug Discov* 2021, **20**:200–216.

110. Saldívar-González FI, Aldas-Bulos VD, Medina-Franco JL, Plisson F: **Natural product drug discovery in the artificial intelligence era**. *Chem Sci* 2022, **13**:1526–1546.

111. Jeon J, Kang S, Kim HU: **Predicting biochemical and physiological effects of natural products from molecular structures using machine learning**. *Nat Prod Rep* 2021, **38**:1954–1966.

112. Bauman KD, Butler KS, Moore BS, Chekan JR: **Genome mining methods to discover bioactive natural products**. *Nat Prod Rep* 2021, **38**:2100–2129.

113. Jarmusch SA, van der Hooft JJJ, Dorrestein PC, Jarmusch AK: **Advancements in capturing and mining mass spectrometry data are transforming natural products research**. *Nat Prod Rep* 2021, **38**:2066–2082.

114. Caesar LK, Montaser R, Keller NP, Kelleher NL: **Metabolomics and genomics in natural products research: complementary tools for targeting new chemical entities**. *Nat Prod Rep* 2021, **38**:2041–2065.

115. Cech NB, Medema MH, Clardy J: **Benefiting from big data in natural products: importance of preserving foundational skills and prioritizing data quality**. *Nat Prod Rep* 2021, **38**: 1947–1953.

116. Haug K, Cochrane K, Nainala VC, Williams M, Chang J, Jayaseelan KV, O'Donovan C: **MetaboLights: a resource evolving in response to the needs of its scientific community**. *Nucleic Acids Res* 2020, **48**:D440–D444.

117. Sud M, Fahy E, Cotter D, Azam K, Vadivelu I, Burant C, Edison A, Fiehn O, Higashi R, Nair KS, *et al.*: **Metabolomics Workbench: an international repository for metabolomics data and metadata, metabolite standards, protocols, tutorials and training, and analysis tools**. *Nucleic Acids Res* 2016, **44**: D463–D470.

118. Alexandrov T, Ovchinnikova K, Palmer A, Kovalev V, Tarasov A, Stuart L, Nigmetzianov R, Fay D, Gaudin M, Lopez CG, *et al.*: **METASPACE: a community-populated knowledge base of spatial metabolomes in health and disease**. *bioRxiv* 2019, https://doi.org/10.1101/539478.

119. Schorn MA, Verhoeven S, Ridder L, Huber F, Acharya DD, Aksenov AA, Aleti G, Moghaddam JA, Aron AT, Aziz S, *et al.*: **A community resource for paired genomic and metabolomic data mining**. *Nat Chem Biol* 2021, **17**:363–368.

120. Walters WP, Barzilay R: **Critical assessment of AI in drug discovery**. *Expet Opin Drug Discov* 2021, **16**:937–947.

121. Huang K, Xiao C, Glass LM, Sun J: **MolTrans: molecular interaction transformer for drug-target interaction prediction**. *Bioinformatics* 2021, **37**:830–836.

122. Wang P, Zheng S, Jiang Y, Li C, Liu J, Wen C, Patronov A, Qian D, Chen H, Yang Y: **Structure-aware multimodal deep learning for drug-protein interaction prediction**. *J Chem Inf Model* 2022, **62**:1308–1317.

123. Nigam A, Pollice R, Krenn M, Gomes GDP, Aspuru-Guzik A: **Beyond generative models: superfast traversal, optimization, novelty, exploration and discovery (STONED) algorithm for molecules using SELFIES**. *Chem Sci* 2021, **12**:7079–7090.

124. Sorokina M, Steinbeck C: **Review on natural products databases: where to find data in 2020**. *J Cheminf* 2020, **12**:20.

125. Rutz A, Sorokina M, Galgonek J, Mietchen D, Willighagen E, Gaudry A, Graham JG, Stephan R, Page R, Vondrášek J, *et al.*: **The LOTUS initiative for open knowledge management in natural products research**. *Elife* 2022:11.

126. Artrith N, Butler KT, Coudert F-X, Han S, Isayev O, Jain A, Walsh A: **Best practices in machine learning for chemistry**. *Nat Chem* 2021, **13**:505–508.

127. UniProt Consortium: **UniProt: the universal protein knowledgebase in 2021**. *Nucleic Acids Res* 2021, **49**:D480–D489.

128. Burley SK, Bhikadiya C, Bi C, Bittrich S, Chen L, Crichlow GV, Christie CH, Dalenberg K, Di Costanzo L, Duarte JM, *et al.*: **RCSB Protein Data Bank: powerful new tools for exploring 3D structures of biological macromolecules for basic and applied research and education in fundamental biology, biomedicine, biotechnology, bioengineering and energy sciences**. *Nucleic Acids Res* 2021, **49**:D437–D451.

2.5 Publication E: DECIMER.ai - An open platform for automated optical chemical structure identification, segmentation and recognition in scientific publications

Rajan, K.[1], Brinkhaus, H.O.[2], Agea, M.A.[3], Zielesny, A.[4], Steinbeck, C.[5]

Submitted to Nat. Commun.: 07th March 2023

Preprint published on Chemrxiv: 23rd February 2023

DOI: 10.26434/chemrxiv-2023-xhcx9

Table 6: Author contributions for Publication E

Author No	**1**	**2**	**3**	**4**	**5**
Conceptual research design	x	x	x	x	x
Planning of research activities	x	x	x	x	x
Tool development (Back-end)	x	x	x		
Web tool development (Front-end)		x			
Data collection	x	x	x		
Data analysis and interpretation	x	x	x		
Manuscript writing	x	x	x	x	x
Suggested publication equivalence value		1.0			

***underlined author numbers refer to involved doctoral students**

DECIMER.ai - An open platform for automated optical chemical structure identification, segmentation and recognition in scientific publications

Kohulan Rajan[1], Henning Otto Brinkhaus[1], M. Isabel Agea[2], Achim Zielesny[3] and Christoph Steinbeck[1*]

[1] Institute for Inorganic and Analytical Chemistry, Friedrich Schiller University Jena, Lessingstr. 8, 07743 Jena, Germany

[2] Department of Informatics and Chemistry, Faculty of Chemical Technology, University of Chemistry and Technology Prague, Technicka 5, 166 28, Prague, Czech Republic.

[3] Institute for Bioinformatics and Chemoinformatics, Westphalian University of Applied Sciences, August-Schmidt-Ring 10, 45665 Recklinghausen, Germany

[*] Corresponding author: christoph.steinbeck@uni-jena.de

Abstract

The number of publications describing chemical structures has increased steadily over the last decades. However, the majority of published chemical information is currently not available in machine-readable form in public databases. It remains a challenge to automate the process of information extraction in a way that requires less manual intervention - especially the mining of chemical structure depictions. As an open-source platform that leverages recent advancements in deep learning, computer vision, and natural language processing, *DECIMER.ai* (Deep lEarning for Chemical ImagE Recognition) strives to automatically segment, classify, and translate chemical structure depictions from the printed literature. The segmentation and classification tools are the only openly available packages of their kind, and the optical chemical structure recognition (OCSR) core application yields outstanding performance on all benchmark datasets. The source code, the trained models and the datasets developed in this work have been published under permissive licences. An instance of the *DECIMER* web application is available at https://decimer.ai.

Main

Introduction

The availability of chemical information in structured data formats and open databases benefits not only researchers in chemistry itself but also scientific fields using chemical information such as medicine, pharmacy, material science, molecular biology and many more [1]. Although substantial efforts exist to establish research data management infrastructures [2,3] and open databases and repositories [4–7], most chemical information is still exclusively published in human-readable text and image formats in the literature. The manual extraction of information from the chemical literature is a time-consuming and error-prone process [8] that can only yield the large amounts of data needed for deep learning applications, for example, when considerable amounts of human resources are employed.

The translation of images containing chemical structure depictions into machine-readable representations is referred to as Optical Chemical Structure Recognition (OCSR). In the last three decades, there has been continuous development in OCSR tools [9,10], most of them being proprietary algorithms [11] and rule-based tools [12–14]. In general, rule-based tools work better with clean images, whereas slight distortions may hinder their performance [15]. In recent years, deep-learning-based OCSR tools have been developed [16,17,18] in conjunction with remarkable advancements in computer vision and natural language processing [19,20]. While several publications have claimed to have developed tools that are capable of recognizing chemical depictions with high accuracy, most of these tools are either proprietary or entirely unavailable [16,21–23]. Among the few open-source OCSR software solutions [15,24], there is no system that combines chemical structure image segmentation, classification, and translation within a comprehensive workflow.

DECIMER.ai, an open-source platform for the identification, segmentation and recognition of chemical structure depictions in the scientific literature, seeks to address this shortcoming. The system combines *DECIMER Segmentation*, a toolkit based on Mask R-CNN [25] for the detection and segmentation of chemical structures in the scientific literature [26], *DECIMER Image Classifier* for the identification of images containing a chemical structure, and *DECIMER Image Transformer* as an OCSR engine, which converts a chemical structure depiction into a machine-readable format. DECIMER algorithms do not inherit any hand-picked rules but instead rely solely on the training data to predict accurate results without making any further hard-coded assumptions.

All components are openly available on GitHub and can be used separately as Python packages or in the user interface of our browser application. The web application is hosted at https://decimer.ai. As all the source code has been published under a permissive licence along with the documentation, users can easily modify and redistribute it or integrate it into their own applications. The Python packages are all hosted on PyPI and are designed to be installable and usable with few lines of code. The *DECIMER.ai* web application can easily be deployed and scaled. As *DECIMER* is trained on publicly available data and is made available to the public in the form of a ready-to-use open-source tool, we believe that the system will significantly reduce the workload and produce high-quality data for the research community and those who are developing and curating chemical databases.

Results

DECIMER Image Classifier and *DECIMER Image Transformer* have been developed and combined with *DECIMER Segmentation* [26] to achieve a comprehensive workflow for the automated extraction and interpretation of chemical structures in the scientific literature (Figure 1). The complete workflow combining all these components is available as a web application with a user interface [27].

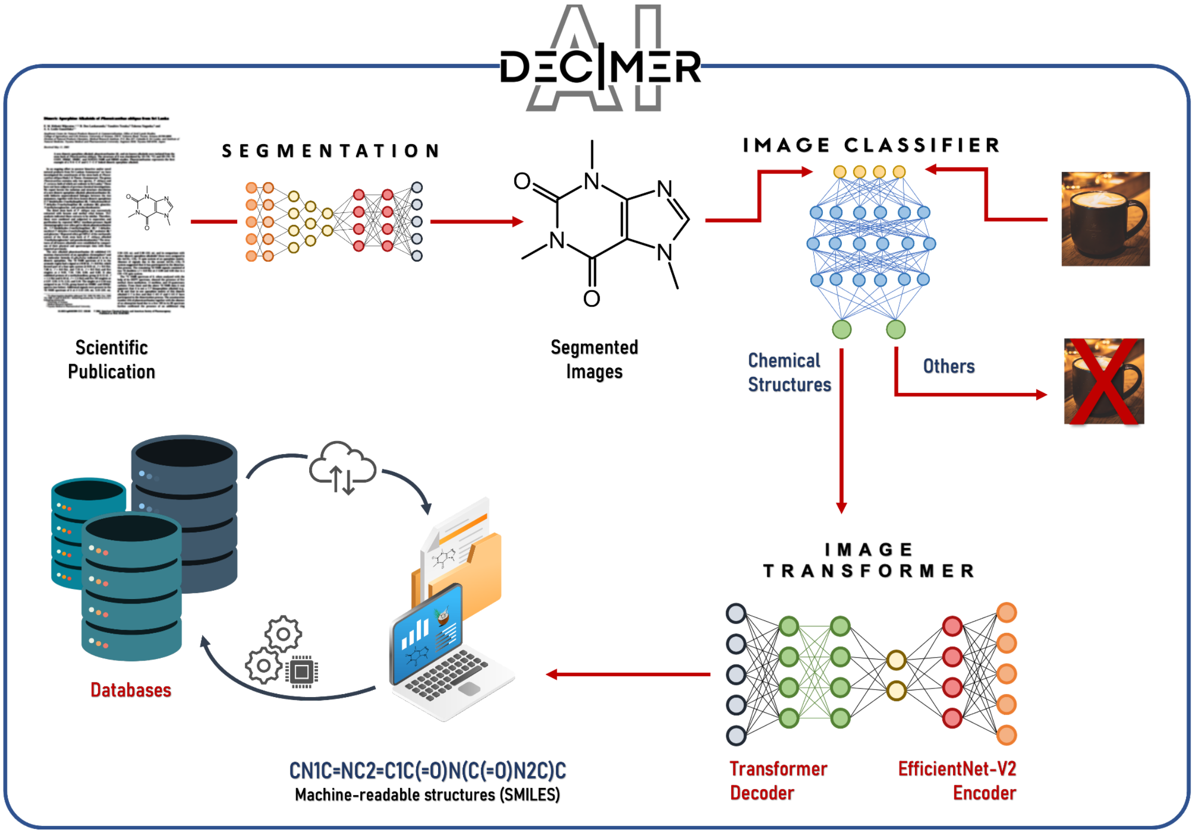


Figure 1: Overview of the integrated *DECIMER* workflow: Detection, segmentation and interpretation of chemical structure depictions in the scientific literature.

DECIMER Image Transformer yields the highest percentage of correct predictions as well as the highest average molecular (Tanimoto) similarities out of all tested tools in our benchmarks (Figure 3). For chemical structure depictions, *DECIMER Image Classifier* is the first openly available classification system and *DECIMER Segmentation* [26] is the only openly available segmentation application. The *DECIMER* web application is the only open-source system that combines these functions in a comprehensive chemical data extraction system.

DECIMER Image Transformer

The key component of *DECIMER.ai* is the *DECIMER Image Transformer* OCSR tool. Due to the usage of diverse chemical structures with diverse depiction features in the training data and an exhaustive image augmentation strategy, the application yields robust results and is capable of interpreting Markush structures as well as common functional groups and superatom abbreviations. A detailed description of the model architecture and the training data is given in the *Methods* section below.

In-domain test performance

The *DECIMER Image Transformer* model was trained with more than 450 million depictions of chemical structures with an image resolution of 512 x 512 pixels (see dataset *pubchem_3* in Supplementary Table 1). The images were generated using the full range of depiction options in the cheminformatics toolkits Chemistry Development Kit (CDK) [28], RDKit [29], Indigo [30] and the Python-based Informatics Kit for Chemical Units (PIKAChU) [31]. A detailed description of the dataset creation can be found in the *Methods* section below.

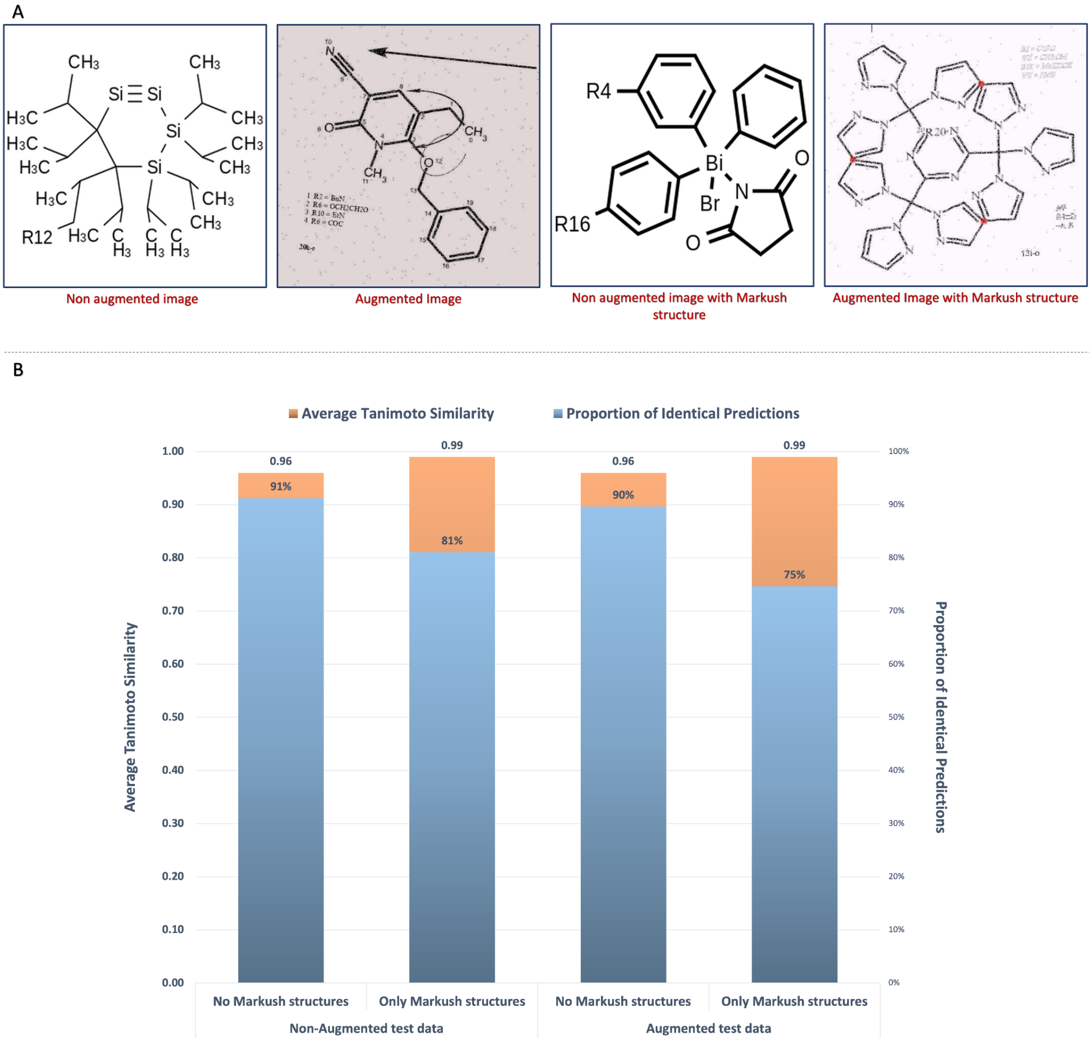
The trained model was tested on four different in-domain datasets containing 250,000 images each. These test datasets were generated similarly to the training datasets but contained no molecules from the training data. In the test datasets, molecules with or without Markush structures or augmentations were included (Figure 2A).

For performance evaluation, two different measures were used: Predictions identical to the correct molecule were considered to be the optimal result, of course. But predictions resembling the correct molecule closely are also very useful for the curation of chemical data. A human curator, for example, who is presented with the bitmap image and an already very similar machine translation only needs to perform a small correction with a chemical structure editor as opposed to re-drawing the whole molecule. To evaluate the similarity of molecular structures, the Tanimoto similarity [32] or Jacard-Index [33] is used, which encodes the presence or absence of structural features of chemical compounds in a bit vector (where PubChem fingerprints were used in particular) and expresses the similarity between two bit vectors (or two chemical structures, respectively) as a number between 0.0 (most dissimilar) and 1.0 (most similar).

In all test results, *DECIMER Image Transformer* consistently produces an average Tanimoto similarity of greater than 0.95. Opposed to the steadily high Tanimoto similarity, there are clear differences regarding the number of perfect predictions. The proportion of perfectly predicted molecules decreases with an increased level of complexity and noise in the structure depictions as well as a lower image resolution.

There are two obvious trends: 1) The addition of image augmentations leads to a lower proportion of perfectly recognised structures. 2) The proportion of perfectly recognised molecules is lower when processing test datasets that exclusively contain Markush structures. These results are not surprising since the R group indices (as in 'R_1') and other labels can be difficult to recognize, especially when the image resolution is low or when additional noise is introduced. Nevertheless, the constantly high Tanimoto similarities indicate that the predicted molecules are very similar to the depicted ones, even when the predictions are not perfect.

Since PubChem fingerprints cannot describe the R-group variables in Markush structures, the derived Tanimoto similarities only describe the similarities of the molecular structures, but cannot be used to evaluate whether the R-group labels have been correctly interpreted. Therefore, the BLEU score [34] was determined as a token-based string similarity metric (see Supplementary Table 2). The obtained average BLEU-score of 0.94 across all test results with Markush structures also indicates a high similarity between the predicted and the true string representations.



Non augmented image

Augmented Image

Non augmented image with Markush structure

Augmented Image with Markush structure

Figure 2: A: Representation of types of images in the training and the test datasets. **B:** In-domain test results: The training dataset includes depictions of Markush structures and a variety of image augmentations (dataset *pubchem_3* in Supplementary Table 1). In the test datasets, these features were separately evaluated to assess their influence on performance. All in-domain test results are also presented in Supplementary Table 2.

OCSR tools benchmark

To assess the performance of the *DECIMER Image Transformer* model in comparison with other openly available tools (OSRA [12], MolVec [14], Imago [13], Img2Mol [15], MolScribe [35], SwinOCSR [24], see Supplementary Table 3), a row of benchmark datasets from a variety of sources was applied (a complete list with additional information about the benchmark datasets and individual tool performance is provided in the *Methods* section and the supplementary information). Following the remark of Clévert et al. [15], that the parameters of the rule-based systems OSRA, MolVec and Imago are overfitted to the available benchmark datasets, mild image distortions (i.e., rotations in the range between -5° and + 5° and mild shearing) were applied to all datasets (see Figure 3B/3D in contrast to Figure 3A/3C for datasets without these distortions).

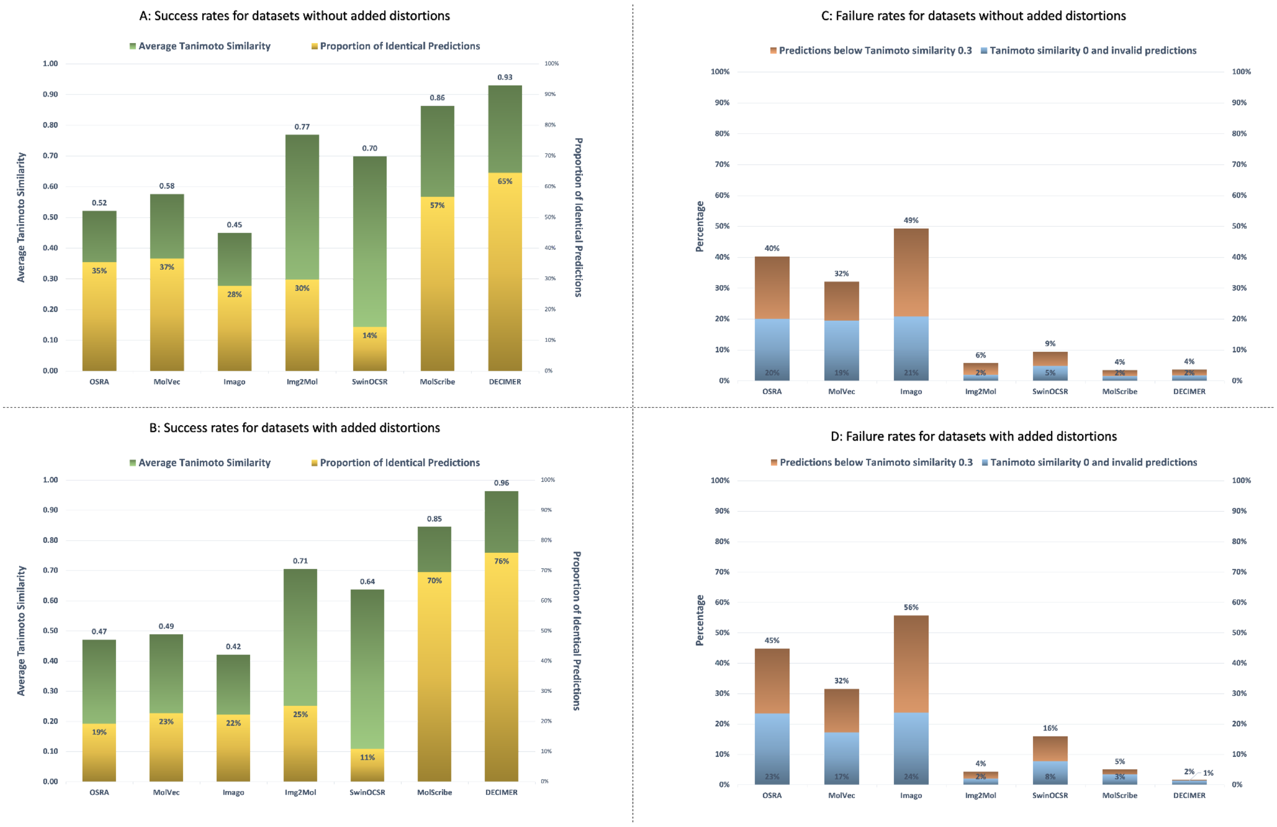


Figure 3: Average performance of the open OCSR tools on all benchmark datasets. The success rates are described by the proportion of perfect predictions and the average Tanimoto similarities, whereas the failure rates are measured as the percentage of predictions with zero Tanimoto similarity plus invalid predictions (catastrophic) and the percentage of predictions with a low Tanimoto similarity value less than or equal to 0.3 (severe). **A:** Success rates for datasets without added distortions. **B:** Success rates for datasets with added distortions. **C:** Failure rates for datasets without added distortions. **D:** Failure rates for datasets with added distortions. The detailed performance metrics for every tool on every benchmark dataset are presented in Supplementary Tables 3 and 4.

DECIMER Image Transformer achieves competitive results on most benchmark datasets compared to the other open OCSR tools (Figure 3), showing no performance degradation due to slight image distortions while confirming the lack of distortion robustness of the rule-based systems. In addition, the rule-based systems fail to correctly recognise the structure depictions with a low image resolution (see *USPTO_big* and *Indigo* in Supplementary Table 3). The

SwinOCSR model does not achieve any outstanding results in our benchmarks - but it needs to be mentioned that its developers stated that their model does not perform well on real-world data [24] which is likely due to a lack of diversity in their training data. Appropriate assessment of failure rates is of particular importance for machine learning applications (Figure 3C/3D): in line with Img2Mol and MolScribe, *DECIMER Image Transformer* exhibits extremely low rates of severe and catastrophic failures.

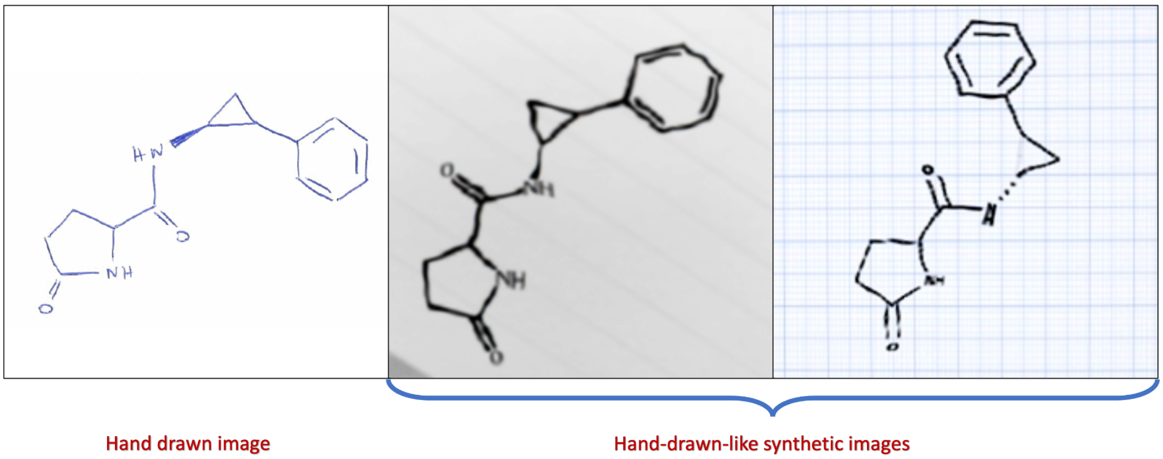


Figure 4: A hand-drawn molecule representation from the *DECIMER Hand-drawn* image dataset [36] (PubChem ID: 31743, left) and corresponding synthetic hand-drawn-like images created with RanDepict [37] (middle, right).

The *DECIMER Image Transformer* model has never been trained on hand-written chemical structure depictions. However, for a benchmark dataset that only consists of hand-drawn chemical structures (*DECIMER Hand-drawn* image dataset outlined in the *Methods* section), it recognises 27% of the structures perfectly and achieves an average Tanimoto similarity of 0.69, whereas all alternative open tools perform worse (see Supplementary Table 3). Moreover, when the model is fine-tuned with a training dataset of images with augmentations that make them appear hand-drawn-like (see Figure 4 and Supplementary Table 1), the proportion of perfect predictions grows significantly to 60% (i.e., an increase of 33%), corresponding to a remarkable average Tanimoto similarity increase of plus 0.2 to 0.89.

DECIMER Image Classifier

DECIMER Image Classifier is a deep learning-based architecture for the identification of images that contain a depiction of a chemical structure. It has been trained, tested and validated using a balanced dataset of images with and without chemical structure depictions (creation and curation of this dataset are outlined in detail in the *Methods* section).

In addition, *DECIMER Image Classifier* has been tested on four external datasets, three publicly available datasets

1. a dataset only containing chemical structure depictions (*ChEBI*),
2. a dataset without any chemical structure depictions (*EM_Images*),
3. a public dataset of images extracted from a diverse set of publications (*PubLayNet*),

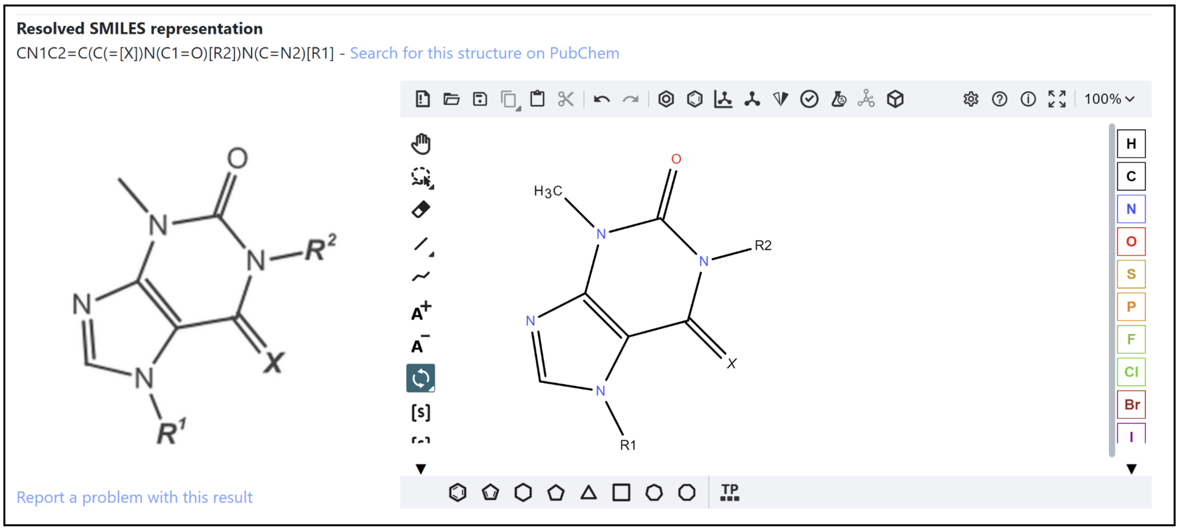
and a manually curated set of images extracted from articles of the Journal of Natural Products (JNP):

4. a real-world dataset using 1000 publications (*JNP_real_world*).

DECIMER Image Classifier predicts a value between 0 and 1, where an optimally determined threshold value is used for the binary decision purpose. The system achieved a 0.99 score on the in-domain test set on every performance metric calculated (Area Under Curve, Matthews Correlation Coefficient, accuracy, specificity, and sensitivity, see *Methods* section below). It correctly classified 99% of the images with chemical structure depictions and almost 100% of images without chemical structure depictions. On the four out-of-domain test sets, the proportion of true classifications was 97% (*ChEBI*), 100% (*EM_Images*), 99% (*PubLayNet*) and 94% (*JNP_real_world*).

DECIMER.ai

DECIMER.ai is a web application that combines the previously described components in an automated, comprehensive workflow for the extraction of chemical structures from the scientific literature. When a user uploads a PDF document or a single image file, *DECIMER Segmentation* is used to cut chemical structure depictions. The segmented chemical structure depictions are then processed by *DECIMER Image Classifier* and *DECIMER Image Transformer* to obtain machine-readable SMILES string representations of the resolved chemical structures.



Resolved SMILES representation

CN1C2=C(C(=[X])N(C1=O)[R2])N(C=N2)[R1] - Search for this structure on PubChem

Report a problem with this result

Figure 5: Example image of a Markush structure (on the left) that has been loaded into the *DECIMER* web application. The SMILES string representation of the molecule is generated (upper left) and depicted in the embedded *Ketcher* molecular editor window (on the right).

These resolved SMILES encoded structures are then automatically loaded in the embedded molecular editor Ketcher [38] (see Figure 5). The molecular editor enables the manual inspection and editing of the resolved chemical structures. In addition to the segmented structure depictions, the resolved structures can be downloaded in the common MOL file format.

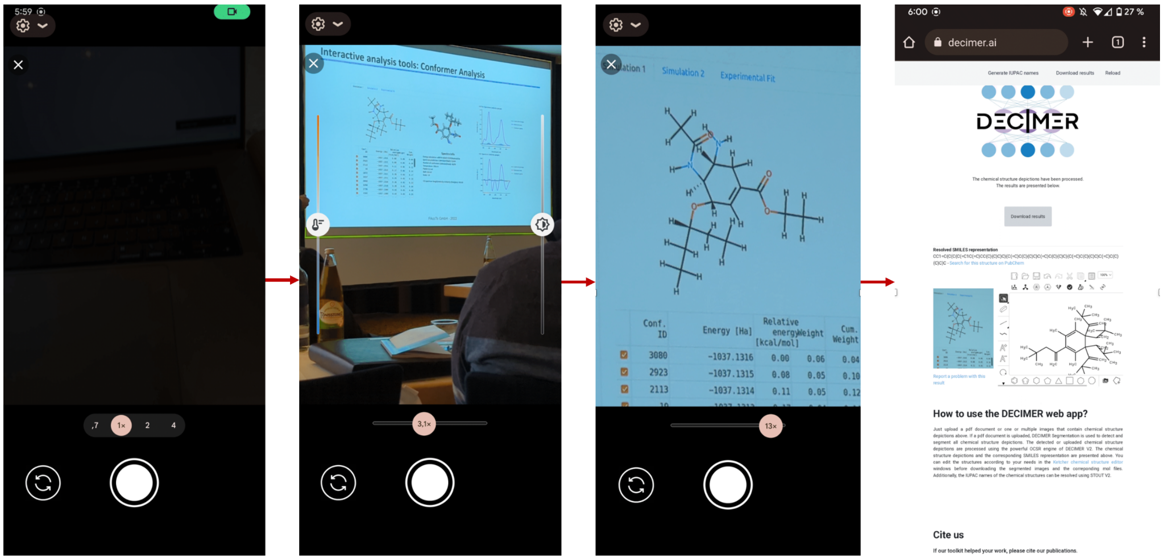
Discussion

DECIMER Image Transformer as the *DECIMER* core component achieves highly accurate results on the in-domain test data. The system performs better on non-augmented test images since augmented images contain a wide range of additional non-structural elements and noise that have to be ignored in order to correctly translate a chemical structure depiction. This effect is diminished when images of a higher resolution are processed because a low-resolution image may already be comparably blurry and may turn unrecognisable when additional augmentations are applied. The *DECIMER Image Transformer* model produces predictions that are highly similar to the original molecules with an average Tanimoto similarity over 0.95. The translation of depictions of Markush structures yields similar results, although the proportion of perfectly predicted structures is considerably lowered. This may be traced to the relevance of the small subscript indices of the R-groups (as in 'R_1'). It could be shown that especially in images with a lower resolution of 299 x 299 pixels, these small digits may become unrecognisable, whereas corresponding images with a resolution of 512 x 512 pixels could be processed with a significant increase in the number of perfectly-recognized Markush structures (see Supplementary Table 2 and Supplementary Figure 1). Moreover, the BLEU scores, which are consistently above 0.9 (see Supplementary Table 2), confirm that the Markush structure predictions are very similar to the original SMILES strings.

In comparison with alternative open OCSR tools, *DECIMER Image Transformer* performs with high accuracy. Apart from *Img2Mol*'s performance on its in-domain test data, MolScribe's performance on USPTO data (which may be part of the system's training data [35]) and the performance of MolVec on the non-distorted JPO, CLEF and USPTO datasets, *DECIMER Image Transformer* performs outstandingly well on all benchmark datasets without any significant differences between non-distorted and distorted images. It is particularly striking that the system's severe and catastrophic failure rates are very low. The system also achieves a comparative peak performance when benchmarked against the *DECIMER Hand-drawn* image dataset, which is especially interesting since there has not been a single hand-drawn structure in the training data. Thus, this model may be applied to extract chemical structures from hand-drawn images in the future. This may become particularly relevant for translating chemical publications from 50 years ago since a lot of the chemical structures from that time were hand-drawn using templates. Although the predictions might not be perfect in all cases, a similar prediction considerably reduces the amount of manual work when mining chemical structures from printed literature. The outlined success rates of *DECIMER Image Transformer* demonstrate not only robust performance, but also superior generalisation capabilities due to a training data diversification strategy with highly diverse structure depictions generated by our OCSR training data generation tool RanDepict [37]: It ensures that the full diversity of depiction features is properly represented that CDK [28], RDKit [29], Indigo [30] and PIKAChU [31] have to offer.

DECIMER Image Classifier is capable of achieving high-performance metrics and is capable of working effectively on a wide range of datasets. In the *ChEBI* dataset, performance was slightly reduced due to the presence of images of isolated ions that were recognized as non-chemical images. None of the electron microscopy images from the *EM_Images* dataset has been wrongly classified as a chemical structure. Considering that the images found in *PubLayNet* originated from diverse sets of articles from PubMed Central, the high performance of the *DECIMER Image Classifier* indicates the robustness of the model. Additionally, the classifier achieved high performance when applied to real-world use cases.

The *DECIMER.ai* web application is the first comprehensive open-source user interface application for the extraction of chemical information from the scientific literature. As discussed above, *DECIMER Image Transformer* translates chemical structure depictions with a high degree of similarity. By embedding it into the *DECIMER.ai* application, a human curator can immediately assess the predictions and correct them in the molecular editor windows if necessary. For the segmentation and classification of chemical structure depictions, *DECIMER Segmentation* and *DECIMER Image Classifier* are the only open-source applications available.



Using the phone camera through *DECIMER.ai*

Zooming in on the image

Capturing the image of the chemical structure

Machine-readable representation of the captured image

Figure 6: *DECIMER.ai* being used via a smartphone at the 17th German Conference on Cheminformatics. The deciphered structure can be searched in PubChem, the largest openly available chemical database, right away.

Since *DECIMER.ai* can be accessed from a mobile phone or tablet via the web browser, these tools are enabled to recognize chemical structures in the real world (Figure 6): By using *DECIMER.ai* on a mobile device during a conference, images of chemical structures may be captured during a presentation or poster session to identify the molecules presented. With the

DECIMER.ai search functionality, users can conduct a direct “single-click” PubChem database search in addition to the structure recognition to access additional chemical information.

There have been closed-source projects like *CLiDE* [39] or the recently published *MolMiner* [23] that combine a segmentation step with an OCSR step in their workflow. *CLiDE* is a fully commercial tool, *MolMiner* permits limited access to registered users and offers unlimited access to users who wish to obtain an enterprise licence. Since the source code of these applications is not openly available, researchers cannot adapt them according to their needs or integrate them into their applications. As all *DECIMER* components and the *DECIMER.ai* web application are open-source projects, continuous further development with significant community-driven improvements can be expected in the future. There have been major advances in the extraction of chemical information from documents. For example, *ChemDataExtractor* [8,40,41] has been used extensively for the automated generation of chemical databases [42–45]. Perspectively, it would be interesting to integrate such applications in *DECIMER.ai* to mine chemical information from the text of PDF documents and link it to structural information obtained from OCSR. Although there are many more challenges to overcome to mine all types of chemical information from the literature using a single platform, *DECIMER.ai* may become a solid open basis for further development.

Methods

The *DECIMER* project was developed as a deep-learning-based solution for OCSR tasks. The goal of the *DECIMER* project is to develop an automated system that detects, segments, and converts images from published literature into computer-readable formats, in this case, the SMILES representation. It is a fully data-driven approach, in which no assumptions are made about the underlying chemical structure. In total, the project is divided into four parts: the segmentation algorithm, the image classifier, the OCSR model, and the web application.

DECIMER Segmentation

Our previously published application *DECIMER Segmentation* [26] was re-used in this work to create a complete extraction workflow. It uses an open implementation of the Mask R-CNN architecture [25] in combination with custom processing steps to segment chemical structure depictions from pages in the scientific literature. Since the original publication, we have refactored the complete codebase, added unit tests and wrapped it up in a Python package that can be installed easily from PyPI [46], but all underlying models and algorithms remain unchanged. The *DECIMER* Segmentation model was trained on manually annotated data using TensorFlow 2.3.0, but it has been updated to work with TensorFlow 2.10.0 in accordance with the other *DECIMER* components. The source code and the model are available on GitHub [47] and Zenodo [48]. For further information about *DECIMER Segmentation*, we would like to refer to the original publication [26].

DECIMER Image Transformer

Selection of molecules

The *DECIMER Image Transformer* model was trained on data based on molecules obtained from PubChem [49]. The entire molecules of PubChem were downloaded in SMILES format directly from the PubChem FTP site [49]. To reduce the imbalance of data, all molecules with a molecular weight of more than 1500 Dalton were filtered out. All explicit hydrogen atoms were removed and stereochemistry was retained. SMILES strings with more than 152 tokens (see *Tokenization*) were filtered out due to their underrepresentation in the data (3,263 molecules). As a result, 108,541,884 molecules were selected in total. A diverse set of 250,000 molecules was selected to use as a test dataset from the whole dataset using the MaxMin [50] algorithm included in chemfp [51]. Another million molecules were selected randomly and used for validation during the development, and the remainder was used as a training dataset (*pubchem_1* see Supplementary Table 1).

Additionally, a second dataset with Markush structures was generated. Due to the unavailability of large datasets of SMILES that represent Markush structures, they were artificially generated based on 20 million SMILES strings which were diversely picked from PubChem [49] using the chemfp [51] implementation of the MaxMin [50] algorithm. To generate SMILES representing Markush structures, the following steps were followed:

1) Read input SMILES using the CDK [28].
2) Add explicit hydrogen atoms and return absolute SMILES.
3) Pseudo-randomly replace 1-3 carbon-'C' or hydrogen-'H' with the rest group variables. Rest group variables are defined as the characters 'R', 'X' and 'Z' with or without an index number between 0 and 20.
4) Read modified SMILES using the CDK.
5) Remove explicit hydrogen atoms and return absolute SMILES.

For example, the input SMILES string 'CCC' is converted to the absolute SMILES string 'C([H])([H])([H])C([H])([H])C([H])([H])[H]'. Subsequently, the pseudo-random insertion of an R-group variable takes place and yields 'C([H])([H])([H])C([H])([H])C([H])([R])[H]'. After re-reading the modified SMILES string and removing the explicit hydrogen atoms, the CDK returns 'CC(C)[R]'. The functionality of generating random Markush structures based on given SMILES strings has been integrated into our open-source OCSR training data generation tool RanDepict [37] for this purpose.

By adding the newly generated SMILES with Markush structures to the SMILES strings from *pubchem_1* and applying the same filtering criteria as described above, 126,702,705 molecules were selected. Based on this, a diverse set of 250,000 SMILES representing molecules with Markush structures were selected for testing using the MaxMin [50] algorithm. One million molecules were retained to use for validation during development, and the remainder was used as training data (*pubchem_2, pubchem_3* see Supplementary Table 1).

Our previous study on the performance of the molecular string representations DeepSMILES [52], SELFIES [53] and SMILES for OCSR purposes [54] with similar model architectures indicates that the usage of SMILES strings leads to more accurate results although the usage of SELFIES leads to more valid chemical structures in the predictions. Thus, SMILES string representations were used for *DECIMER Image Transformer*.

Tokenization

The SMILES strings in the datasets were split into meaningful tokens using the Keras [55] tokenizer with TensorFlow 2.8.0 [56]. The following set of rules was applied where each string is split after,

- every heavy atom: e.g., “C”, “Si”, “Au”
- every open bracket and closed bracket: “(”, “)”, “[”, ”]”
- every bond symbol: “=”, ”#”
- every one of the following characters: “.”, “-”, ”+”, ”\”, ”/”, ”@”, ”%”, ”*”
- every single-digit number

After the splitting, a “<start>” and an “<end>” token were added at the beginning and the end of the sequence. To match the same maximum length, each tokenized string was padded with “<pad>” tokens. The token “<unk>” is used for unknown elements and acts as a placeholder. R-group indices were replaced according to the procedure described in the subsection *Evaluation of different R-group representations in SMILES*.

The following is a list of all tokens found in dataset *pubchem_1*:
<unk>, C, =, (,), O, N, 1, 2, 3, <start>, <end>, @, [,], 4, H, F, S, 5, Cl, /, ., 6, -, +, Br, #, \, 7, 8, 9, P, I, Si, B, Na, K, %, Se, Sn, Y, Li, Zr, Fe, Ti, Al, Zn, Pt, Cu, Ir, Mg, Ni, Co, W, Ru, Ca, Ge, V, As, 0, Pd, Cr, Mn, Sb, Ag, Te, Hg, Mo, Hf, Rh, Au, Pb, Ba, Bi, U, Rb, In, Cs, Ga, Re, Cd, Ar, Sr, Os, Ce, La, Gd, Tl, Nb, Nd, Ta, Eu, Pr, Sm, Yb, Sc, Be, Tb, Dy, Er, Th, Lu, Ho, *, Tm, Xe, He, Pa, Kr, Ne, <pad>

The following is a list of all tokens found in dataset *pubchem_2*:
<unk>, C, =, (,), O, N, 1, 2, [,], 3, <start>, <end>, @, 4, H, F, S, Cl, 5, /, !, X, Z, R, ., 6, Br, +, -, #, \\, §, $, 7, £, <, ?, ¢, ^, >, €, 8, I, P, 9, Si, B, Na, %, Se, 0, Sn, K, Y, Li, Zr, Fe, Al, Ti, Zn, Pt, Cu, Ir, As, Ni, Mg, Ge, W, Co, Ru, Ca, V, Pd, Te, Cr, Mn, Sb, Hg, Ag, Mo, Pb, Hf, Bi, Au, Rh, Ba, U, In, Rb, Ga, Re, Cs, Cd, Sr, Ar, Tl, Ce, Os, La, Nb, Gd, Ta, Nd, Eu, Pr, Sm, Yb, Sc, Be, Tb, Th, Er, Dy, Lu, Ho, *, Tm, Xe, He, Kr, Pa, Ne, <pad>

Generation of chemical structure depictions

The images of chemical structures were depicted as grayscale 2D bitmap images using our open-source toolkit RanDepict [37]. In the chemical literature, various types of chemical structure depictions are represented. This is due to the usage of numerous different software packages or even templates for hand-drawing chemical structures throughout different types of publications. RanDepict attempts to generate datasets in which all features that define different types of depictions are represented in a balanced and controlled manner by pseudo-randomly scrambling all available depiction parameters for every created image. Additionally, a variety of image augmentations like rotation, shearing, the addition of curved arrows in a structure, and the addition of text labels and reaction arrows around the structure can be applied [37].

The originally published version of RanDepict (1.0.5) uses the CDK [28], RDKit [29] and Indigo [30] toolkits to generate diverse sets of chemical structure depictions. For the training dataset *pubchem_1*, this version of RanDepict was used to depict each molecule once without and three times with image augmentations with different pseudo-randomly scrambled depiction parameters for each image.

Since then, we have continued the development of RanDepict and have implemented the option to depict Markush structures. Additionally, the generation of SMILES representations of Markush structures based on any given SMILES string that has been described in the section *Selection of molecules* was implemented. Furthermore, we contributed to PIKAChU [31], to allow the depiction of Markush structures and implemented its functionalities in RanDepict. Finally, RanDepict 1.0.8 was used to generate the chemical structure depictions in the training dataset *pubchem_2*, which contains Markush structures where the images were depicted with a size of 299 x 299 pixels. Here once again, one depiction was created without any augmentations, and three depictions were created with augmentations. This version of RanDepict produced some invalid SMILES representations of Markush structures resulting in a reduction of total images. Due to the large number of depictions (479,500,000 images) and the time and resources spent on their production, we decided to proceed with this dataset.

To evaluate the performance of the model using images with a higher resolution, a third dataset was created by re-depicting the molecules from the *pubchem_2* dataset with an image size of 512 x 512 pixels (where originally the images on *pubchem_2* dataset were depicted with an image size of only 299 x 299 pixels). Everything else was done following the same procedure as the production of *pubchem_2*. During the creation of the dataset, not all molecules were completely depicted due to memory issues, resulting in a reduction in the number of images. Again, we decided to use the generated training dataset since there were more than 453,900,000 million images. This dataset is referred to as *pubchem_3*.

RanDepict version 1.1.4 has been used to generate 127,500,000 hand-drawn-like synthetic structure depictions with an image size of 512 x 512 pixels using the *pubchem_3* dataset. The augmentation functionalities that enable the generation of a hand-drawn-like style that has been implemented in RanDepict for this purpose are based on ChemPIX's implementation of hand-drawn-like hydrocarbon chemical depictions [57].

All training datasets were saved as TFRecord files to enable the training on TPU cloud instances using Tensorflow. Due to the large number of data points used in our training datasets (>400,000,000), the training dataset generation is a time-consuming process. Consequently, the SMILES datasets were divided into 100 chunks of equal length and used as input for the RanDepict toolkit which was instantiated with different seeds to produce different sets of depiction features in each instance. To create TFRecord files directly from SMILES input, a custom Python script was used which is available in the RanDepict repository. The 100 SMILES list chunks per training dataset were processed on an in-house cluster using the workload manager Slurm. In each instance, 20 threads were used on virtual machines with 36 processor cores (2x Intel Xeon Gold 6140 18 Core 2,3 GHz) and 192 GB of RAM. Generating the datasets with an image size of 512 x 512 pixels took almost two weeks.

Model Selection

DECIMER Image Transformer is based on an encoder-decoder architecture. A convolutional neural network (CNN) encoder generates feature vectors from 2D images which are then decoded by a transformer model [58] to yield a SMILES representation of the depicted molecule. The CNN encoder architecture used for *DECIMER Image Transformer* is EfficientNet-V2 [59]. Specifically, the EfficientNet-V2-M CNN model was chosen without any further modifications. A transformer model was used as the decoder. The transformer used in this work has four layers and eight parallel attention heads. The attention has a dimension size of 512 and the feed-forward networks have a dimension size of 2048.

Training

All of the *DECIMER Image Transformer* models were trained on TPUs available on the Google Cloud Platform (GCP). For training models, GCP offers a variety of TPUs. In this work, TPUs were selected for training models primarily due to their faster training speed, scalability, and availability on the Google Cloud Platform. To enable the training on TPU devices, all datasets were saved as TFRecord files.

The training of models that were trained on the datasets *pubchem_1* and *pubchem_2* was run using a TPU V3-32 pod slice. The TPU V3-32 pod slice consists of four devices, which equals 32 nodes in total. This results in a fourfold increase in training speed compared to the previously used TPU V3-8 devices. The model trained using the *pubchem_3* dataset was trained on a TPU V3-256 pod slice.

All models were trained using the Adam optimizer with a custom learning rate scheduler. Sparse categorical cross entropy was used as a loss metric. The dropout rate was set to 0.1 to avoid overfitting. When training models using the images with the size of 512 x 512 pixels, the per-node batch size was set to 48. Training scripts and models are written in Python 3 with Keras and Tensorflow 2.8.0.

Computational considerations

Training a model with the training dataset *pubchem_1* on the TPU V3-8 device took nearly 3 days and 10 hours on average per epoch. Training the same model using a TPU V3-32 pod slice took an average of one day and two hours. Thus, it was decided to train all models on TPU pod slices of V3-32 or higher to speed up the training process.

To train the models using the training dataset *pubchem_3*, the encoder had to be configured to accommodate the larger image size. Three EfficientNet-V2 encoder models were used to train and test the models trained using mages with the size of 512 x 512 pixels. These are EfficientNet-V2-B3, EfficientNet-V2-S, and EfficientNet-V2-M.

The training of the models with EfficientNet-V2-B3 per epoch took an average of 2 days and 3 hours on a TPU pod slice V3-32. All training processes were moved to a TPU pod slice V3-256 to speed up training. Using EfficientNet-V2-B3, a model could be trained within 12 hours and 30 minutes on average per epoch after changing the training device. For the model with EfficientNet-V2-S as the encoder, it took 15 hours and 26 minutes on average to train each epoch, while for the model with EfficientNet-V2-M as the encoder, it required 1 day and 7 hours.

Test datasets

In order to test the model trained on *pubchem_3*, the previously selected set of 250,000 molecules was used. Each of these molecules was depicted twice at 512 x 512 pixels, with and without augmentations. Moreover, to test the effectiveness of the model against images of chemical structures depicted with Markush structures, another dataset of 250,000 molecules, diversely selected using the MaxMin [50] algorithm and depicted using RanDepict, was used, where the images were shown both with and without augmentation at a resolution of 512 x 512 pixels.

Evaluation of the test results

The analysis of the test results was conducted using metrics generated with the CDK. All predicted SMILES strings for the test datasets were parsed using the CDK SMILES parser. Those that did not get parsed were labelled as invalid SMILES, whereas those that did get parsed were labelled as valid SMILES strings. Using the valid SMILES strings, accuracy and similarity were calculated by comparing each predicted SMILES string with the original SMILES string.

We initially generated InChI strings from the original and predicted molecules and compared them one to one in order to determine the accuracy of the model. For models trained using images with R-Group labels, obtaining InChI strings to calculate identical string matches is not possible. In order to overcome this problem, all of the original and predicted SMILES were parsed using the CDK SMILES parser, and an Isomeric CX SMILES was generated by combining CDK's Absolute and CXSMILES flavours. The SMILES string generated using this method then consists of a canonicalised SMILES with a ‘*’ symbol where the R-Group should be present. At the end of each SMILES string, the R-Group labels that need to be inserted for the asterisks are listed. Using this particular SMILES variant, a one-to-one string comparison was performed, to determine the proportion of identical predictions.

Considering that the *DECIMER Image Transformer* model could potentially predict similar, but not identical molecules, it is important to also examine the similarity of the predicted molecules. Each predicted and original SMILES string pair was converted into CDK’s iAtomContainer objects, and a Tanimoto similarity index was calculated based on PubChem fingerprints for each pair of original and predicted structures.

After all, the metrics have been calculated for each pair in the test dataset. The proportion of valid SMILES predictions, invalid SMILES predictions, the average accuracy, the average Tanimoto index and the proportion of Tanimoto 1.0 occurrences were calculated for each test dataset.

The BLEU (bilingual evaluation understudy) scores were calculated in addition to determining the accuracy of the predictions made by the model that predicts SMILES for images with Markush structures. This score evaluates how well a model can predict SMILES that are similar to the original molecule's SMILES.

Evaluation of different R-group representations in SMILES

Many Markush structures have more than one R-group attached to them. Therefore, the R-groups are commonly assigned indices, as in ‘R_1’ or ‘R_2’. When creating SMILES strings with R-group representations, this leads to the introduction of tokens with multiple meanings. For example, the token ‘1’ in the SMILES string ‘c1ccccc1[R1]’ can represent a ring opening or closure or an R-group index. To evaluate the influence of this potential problem, the performance of two models was compared.

In order to assess if these different possibilities of interpretation of the same tokens have an impact on the performance, two models were trained and tested. The first model was trained on images with R-group depictions and SMILES strings with R-group labels without any further modifications. The second model has trained on the same images, but the matched SMILES strings were modified to avoid tokens with multiple meanings. Every digit that occurs right after an R-group label is replaced by a character that does not have any function in the SMILES syntax. The following replacement characters were used:

1 → !, 2 → $, 3 → ^, 4 → <, 5 → >, 6 → ?, 7 → £, 8 → ¢, 9 → €, 0 → §

For example, this converts the SMILES string ‘C[R5]N1C=NC2=C1C(=O)N(C(=O)N2C)C[R12]’ into ‘C[R>]N1C=NC2=C1C(=O)N(C(=O)N2C)C[R!$]’.

The SMILES representations of Markush structures were downloaded from the SwinOCSR [24] repository and the images were generated using the CDK depiction generator with a resolution of 299 x 299 pixels. Character-based tokenisation was applied in both cases. Both models were trained on a set of 1 million structure depictions and tested on a set of 102,400 molecules from the whole dataset (selected using the MaxMin [50] algorithm) which were depicted as separate images; these molecules were not included in the training data.

For the evaluation, the original digits were re-inserted into the SMILES strings predicted by the second model. The SMILES strings were canonicalised and the Tanimoto similarity based on PubChem fingerprints was computed using the CDK. The performance evaluation was done based on the average Tanimoto similarity, the proportion of Tanimoto similarity values of 1.0, the proportion of exact string matches based on the canonical SMILES and the proportion of valid predicted SMILES representations of molecules (Figure 7).

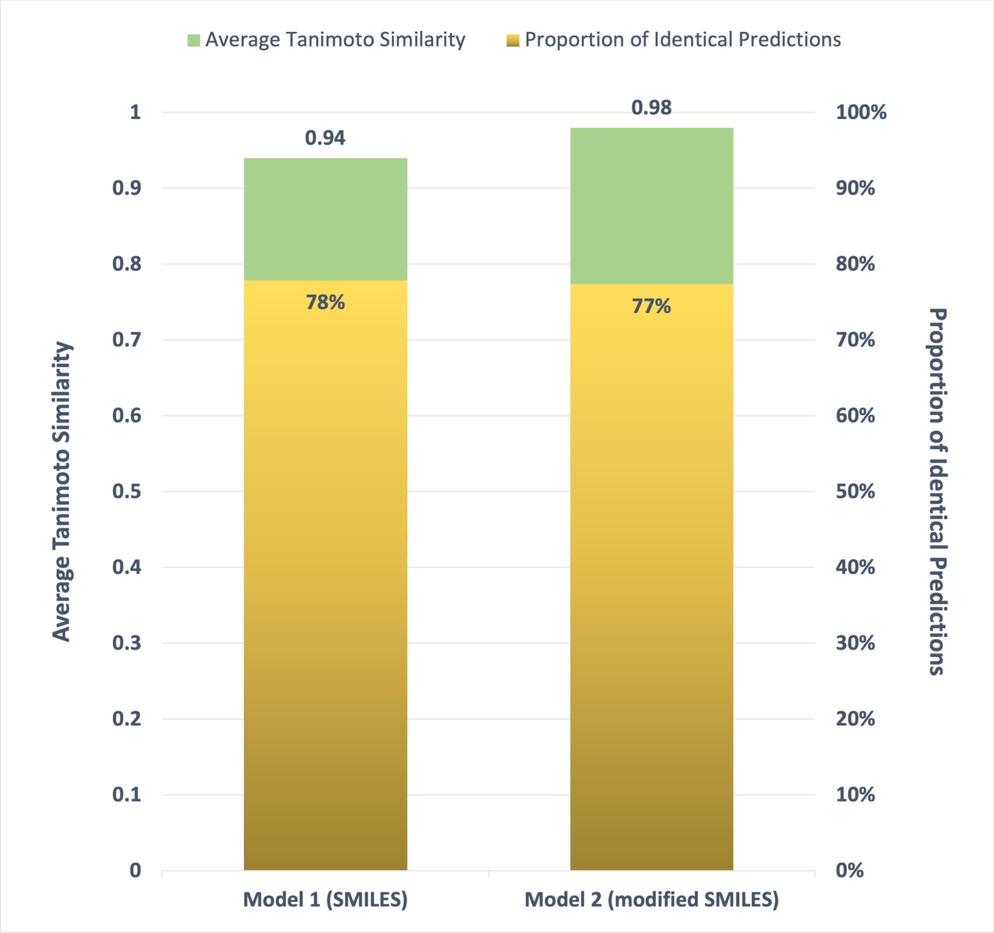


Figure 7: Test performance of a model trained on SMILES strings without further modifications (Model 1) and SMILES strings with replaced R-group indices (Model 2)

The model that was trained on the modified SMILES representation of Markush structures outperforms the model that was trained on the original SMILES representations. It yields a higher proportion of valid predicted SMILES strings (+3.4%), a higher proportion of Tanimoto 1.0 similarities (+2.2%) and a higher average Tanimoto similarity (+0.04), although the number of perfect predictions is slightly worse (-0.5%) for more details see Supplementary Table 5.

The results show that the double meaning of tokens in the training data (digits as part of the ring syntax or as an R-group index) leads to a lower performance of the model trained with it. Based on this finding, the modified SMILES representations were used for the training of all *DECIMER Image Transformer* models described in this publication.

Benchmark

We determined the performance of the *DECIMER Image Transformer* and other available OCSR tools to assess their ability to be applied in a real-world use case to automate the mining of chemical structure depictions from the printed literature. A comprehensive benchmark of the *DECIMER Image Transformer* was conducted using all publicly available OCSR benchmark datasets and *DECIMER* test datasets.

The first four datasets were downloaded from *Rajan et al.* OCSR Review GitHub Page [60]. The other ones were generated or downloaded from the noted sources.

- *USPTO*: A set of 5,719 images of chemical structures and the corresponding MOL files (US Patent Office) obtained from the OSRA online presence [61]

- *UOB*: The dataset of 5,740 images and MOL files of chemical structures developed by the University of Birmingham, United Kingdom, and published alongside MolRec [62]

- *CLEF*: The Conference and Labs of the Evaluation Forum test set of 992 images and molfiles published in 2012 [63]

- *JPO*: A subset (450 images and MOL files) of a dataset based on data from the Japanese Patent Office, obtained from the OSRA online presence [61]. Note that this dataset contains many labels (sometimes with Japanese characters) and irregular features, such as variations in the line thickness. Additionally, some images have poor quality and contain a lot of noise.

- *RanDepict250k*: A set of 250,000 chemical structure depictions generated with RanDepict (1.0.8) using RanDepict's depiction feature fingerprints [37] to ensure diverse depiction parameters. None of the depicted molecules is present in the *DECIMER* training data. The images here are all 299 x 299 pixels in size.

- *RanDepict250k_augmented*: A set of the same 250,000 images from the RanDepict250k dataset. Additional augmentations (examples: mild rotation, shearing, insertion of labels and reaction arrows around the structures, insertion of curved arrows in the structure) were added to the images using RanDepict. The images here are all 299 x 299 pixels in size.

- *DECIMER hand-drawn* [36]: A set of 5,088 chemical structure depictions which were manually drawn by a group of 24 volunteers. The drawn molecules have been picked using the MaxMin [50] algorithm from all molecules in PubChem [49] so that the set represents a big part of the chemical space.

- *Indigo*: 50,000 images generated by Staker et al. [16] using Indigo[30] which were collected from the supplementary information. All images have a resolution of 224 x 224 pixels.

- *USPTO_big*: 50,000 images from the USPTO from Staker et al. [16] which were collected from the supplementary information. All images have a resolution of 224 x 224 pixels.

- *Img2Mol test set*: A set of 25,000 chemical structure depictions used by Clévert et al. for testing [15]. All images have a resolution of 224 x 224 pixels.

DECIMER Image Transformer was also benchmarked against a set of distorted datasets. These images were generated using the original OCSR benchmark datasets, but with a slight shearing and rotation. The *Img2Mol* and *DECIMER hand-drawn* images datasets were not perturbed because they already contained a mixture of clean and perturbed images.

The following paragraphs describe the steps that were taken to run all the openly available OCSR tools.

The compilation of OSRA with all of its dependencies is a complex task. To facilitate the usage, we have modified a version of docker-osra [64], a dockerised version of OSRA to update it to the newest version (at the time of publication: OSRA 2.1.3). The docker image of the version we used is available DockerHub [65]. To use it on our high-performance computing (HPC) cluster, the Docker image has been run with Singularity, an open-source containerisation application.

```
singularity run --bind /root_path/ docker://obrink/osra:2.1.3 sh
/root_path/scripts/run_osra_batch.sh /root_path/input_image/dir
/root_path/output_sdfile_path
```

The command above runs the script run_osra_batch.sh in the Docker image using Singularity. The script runs OSRA on every image in a given directory and saves the resolved structure as an SD file in a second given directory.

Content of run_osra_batch.sh:

```
#!/bin/bash

for image in $1/*.png;
    do echo $image && osra -f sdf -w $2/${image##*/}.sdf $image;
    done;
```

MolVec was downloaded as a jar file containing all dependencies [66]. It was used by running

```
java -jar /path/to/molvec-0.9.8-jar-with-dependencies.jar -dir
/path/of/input/image_dir/ -outDir /path/of/output/molfile/dir
```

Imago 2.0.0 was used via its command line utility with the compiled executable provided by the developer epam [67].

```
imago_console -dir /path/of/input/image_dir/
```

Img2Mol uses an encoder-decoder architecture. The original version of Img2Mol relies on an HTTP request of the encoded image to a server hosted by Bayer where the decoder is running. As the web server is only meant to be used for demonstration purposes, we contributed to Img2Mol to create a version that runs the decoder locally instead of sending HTTP requests to server [68]. This standalone version has been used to process all available benchmark datasets by running. The content of the script img2mol_batch_run.py is given in the supporting information (see Code Resource 1 in the supplementary information).

```
python img2mol_batch_run.py /input/path/ output/path png
```

As the original version of SwinOCSR did not include an inference script, we contributed to the open-source project to facilitate the usage of the model with the best performance according to the authors (focal loss model) [24]. After cloning the repository [69] and preparing the environment according to the instructions given there, it was used by running the following command in the directory that contains the scripts related to the above-mentioned model in the repository (SwinOCSR/model/Swin-transformer-focalloss/):

```
python run_ocsr_on_images.py --data-path /path/to/directory/with/images/
```

DECIMER Image Classifier

Generation of chemical structure depictions

Chemical structures were depicted as PNG images using the open-source toolkit RanDepict [37]. Five different chemical structure depictions were generated for each entry in the ChEMBL30 [70] database (2,157,379 compounds) and the COCONUT database [71] (407,270 natural products). Once the chemical structure depictions were generated, the number of images without chemical structures was determined (6,814,929). In the next step, the same number of images with chemical structures was randomly selected. Following the selection of images with chemical structures, the dataset was randomly divided into training, validation, and test sets based on the 80:16:4 ratio. The result was a training data set containing 5,452,557 structure depictions, a validation data set containing 1,089,899 depictions, and a test data set containing 272,473 depictions.

Generation and assembly of images without chemical structure depictions

Using the matplotlib package in Python, 404,597 images of random graphs were generated with various options concerning plotting style, background, and text size. Additionally, we selected datasets containing images that could be mistaken for chemical structures, that could be easily presented in scientific papers or other diverse datasets (see Supplementary Table 6). In total, 6,410,332 images were retrieved from the public domain; a complete list of the datasets used can be found in Supplementary Information Table 6. In the same manner as the chemical images, the images with non-chemical data were randomly divided into training, validation, and test sets following a 80:16:4 ratio.

Preparation and training

DECIMER Image Classifier is based on the EfficientNet-V1-B0 model and was fine-tuned using 10,905,114 images, validated on 2,179,798 images, and tested on 544,946 images. The images were split randomly. With a batch size of 650 and five augmentations (vertical and horizontal flips, rotations, contrasts, and zooms), the whole training and validation process took about 52 hours and 15 minutes using a Tesla V100-PCI-E-32GB GPU.

Performance evaluation

The performance of the *DECIMER Image Classifier* was determined by evaluating its predictions on the test dataset. Initially, the Area Under the Curve (AUC) which measures the probability of correctly identifying instances more frequently than at random was calculated. Based on a value range from 0 to 1, 1 indicates an accurate classification, and 0.5 indicates total randomness. The *DECIMER* model's AUC for the test set is 0.99 (Supplementary Information Figure 2).

The AUC allows the calculation of the highest distance between the curve and the random prediction. This is referred to as the Youden index (J) [72,73] and it reflects the model's threshold that achieves the best separation between the classes (chemical structure or no chemical structure).

Having established the most appropriate classifier threshold, other performance metrics using the confusion matrix can be computed, which include True Positive (TP), True Negative (TN), False Positive (FP), and False Negative (FN) values:

- Sensitivity = $\frac{TP}{TP + FN}$; known as the true positive rate; the higher the score, the higher the proportion of TP in the set of positive predictions
- Specificity = $\frac{TN}{TN + FP}$; known as the true negative rate; the higher score, the higher the proportion of TN in the set of negative predictions.
- Matthews Correlation Coefficient (MCC) = $\frac{TP\,x\,TN - FP\,x\,FN}{\sqrt{(TP + FP)(TP + FN)(TN + FP)(TN + FN)}}$. The MCC is ranked between -1 and 1, where 1 represents a perfect classification while 0 represents a complete random sample.

- Accuracy = $\frac{TP + TN}{TP + FN + TN + FP}$. This is the proportion of correct predictions for both classes.

Test datasets

The *DECIMER Image Classifier* was tested using four different datasets:

- *ChEBI* (Chemical Entities of Biological Interest) [74] database: This database was filtered to exclude structures found in ChEMBL and COCONUT databases, and five diverse depictions of each molecule were created using RanDepict, resulting in a total of 416,925 images.
- *EM_Images* (from Kaggle): This dataset contains 49,684 images of electron microscopy.
- *PubLayNet* [75]: This collection consists of 57,492 images illustrating figures from printed literature.
- *JNP_real_world*: A set of 8,733 images that were automatically segmented from 1,000 publications from the Journal of Natural (JNP) products using *DECIMER Segmentation*. The segments were manually inspected by a human curator.

DECIMER.ai

The *DECIMER.ai* web application has been developed using Laravel 8, a PHP-based web framework that follows the model-view-controller (MVC) design pattern. It runs as a three-container Docker application that can be deployed using docker-compose. The three containers are responsible for running the nginx web server, communicating between the user interface and the processes running in the background and managing the deep-learning applications in the background.

When the app is launched, a user-defined number of socket server instances is started. Each of these socket servers listens to a different local port and waits to receive the location of an image to process. Multiple instances of each model type can be loaded. Working with multiple instances of these local socket servers allows fast parallel processing at the cost of constant memory usage for the preloaded models. This procedure was chosen to ensure a pleasant and fast user experience without the need to reload the models at every processing step.

Once the user uploads a PDF document, it is converted to multiple image files (one per page). The locations of these image files are then distributed over all available socket servers that run a preloaded model instance of *DECIMER Segmentation*. Once the chemical structures have been detected, the images are saved and their locations are sent back to the user interface where they are displayed. In parallel, the locations of the segments are sent to all available socket server instances that run preloaded instances of the models of *DECIMER Image Classifier* and *DECIMER Image Transformer*. The classifier instances receive the image path and return the values ‘true’ or ‘false’ based on whether the image is classified as a chemical structure depiction or not. The *DECIMER Image Transformer* instances receive an image path and return a resolved SMILES string. At this point, based on the SMILES strings, the corresponding molecules are displayed in the embedded *Ketcher* molecular editor [38] windows in the user interface and a warning is displayed if the image is not classified as a chemical structure depiction. Then, the user can download the segmented structure depictions, the corresponding MOL files and a file with the SMILES representations. If a single image is directly uploaded instead of a PDF, the same procedure of segmentation and subsequent OCSR processing is followed. If multiple images are uploaded, their locations are directly sent to the *Image Transformer* instances. If the user hits the button on the user interface, the resolved SMILES strings are sent to the STOUT [76] socket server instances, which return the corresponding resolved IUPAC names.

An instance of the *DECIMER* web application is available at https://decimer.ai. The complete source code is openly available on GitHub at https://github.com/OBrink/DECIMER.ai. The GitHub repository contains instructions on how to set up the web app locally and how to scale the memory requirements (as well as the parallel processing speed) by changing the number of socket servers with preloaded model instances.

Data availability

The datasets used for *DECIMER Image Transformer* were directly retrieved from PubChem https://ftp.ncbi.nlm.nih.gov/pubchem/Compound/Extras/CID-SMILES.gz

The *DECIMER Image Classifier* dataset: https://zenodo.org/record/6670746

The trained checkpoints available at:

- DECIMER Segmentation: https://doi.org/10.5281/zenodo.7299334
- DECIMER Classification: https://github.com/Iagea/DECIMER-Image-Classifier/tree/main/decimer_image_classifier/model
- DECIMER Image Transformer: https://doi.org/10.5281/zenodo.7624994

Code availability

The *DECIMER Segmentation* algorithm is available at:
https://github.com/Kohulan/DECIMER-Image-Segmentation
The *DECIMER Image Classifier* is available at:
https://github.com/Iagea/DECIMER-Image-Classifier
The *DECIMER Image Transformer* is available at:
https://github.com/Kohulan/DECIMER-Image_Transformer
The *DECIMER.ai* code is available at:
https://github.com/OBrink/DECIMER.ai

List of abbreviations

AUC - Area Under the Curve
BLEU - BilinguaL Evaluation Understudy
CDK - Chemistry Development Kit
ChEBI - Chemical Entities of Biological Interest
CLEF - Conference and labs of the evaluation forums
CLIDE - Chemical literature data extraction
CNN - Convolutional Neural Networks
COCONUT - COlleCtion of Open Natural prodUcTs
CXSMILES - ChemAxon Extended SMILES
DECIMER - Deep lEarning for Chemical ImagE Recognition
FTP - File Transfer Protocol
FN - False Negative
FP - False Positive
GCP - Google Cloud Platform
GPU - Graphical Processing Unit
InChI - International Chemical Identifier
J - Youden index
JNP - Journal of Natural Products
JPO - Japanese Patent Office
MCC - Matthews Correlation Coefficient
NP - Natural Products
OCSR - Optical Chemical Structure Recognition
OSRA - Optical Structure Recognition Application
PIKAChU - Python-based Informatics Kit for Analysing CHemical Units
PDF - Portable Document Format
PNG - Portable Network Graphics
PyPI - Python Package Index
R-CNN - Region-Based Convolutional Neural Networks
R-group - Rest group
ROC - Receiver Operating Characteristic
SELFIES - Self-referencing embedded strings
SMILES - Simplified Molecular-Input Line-Entry System
STOUT - SMILES-TO-IUPAC-name Translator
TFRecord - TensorFlow Record file
TN - True Negative
TP - True Positive
TPU - Tensor Processing Unit
UOB - University of Birmingham, United Kingdom (dataset)
USPTO - United States Patent and Trademark Office
VM - Virtual Machine

Acknowledgements

We acknowledge Dr. Christoph Riplinger, FAccTs GmbH, Cologne, Germany, for the permission to use photos of this slides shown in figure 6.
We are grateful for the company Google making free computing time on their TPU Research Cloud infrastructure available to us and KR acknowledges the research supported with Cloud TPUs from Google's TPU Research Cloud (TRC).

Author information

Kohulan Rajan
Present Address: Institute for Inorganic and Analytical Chemistry, Friedrich Schiller University Jena, Lessingstr. 8, 07743 Jena, Germany

Henning Otto Brinkhaus
Present Address: Institute for Inorganic and Analytical Chemistry, Friedrich Schiller University Jena, Lessingstr. 8, 07743 Jena, Germany

M. Isabel Agea
Present Address: Department of Informatics and Chemistry, Faculty of Chemical Technology, University of Chemistry and Technology Prague, Technicka 5, 166 28, Prague, Czech Republic

Achim Zielesny
Present Address: Institute for Bioinformatics and Chemoinformatics, Westphalian University of Applied Sciences, August-Schmidt-Ring 10, 45665 Recklinghausen, Germany

Christoph Steinbeck
Present Address: Institute for Inorganic and Analytical Chemistry, Friedrich Schiller University Jena, Lessingstr. 8, 07743 Jena, Germany

Corresponding author: christoph.steinbeck@uni-jena.de

Authors' contributions

KR & HOB developed the software suite and performed the analysis. MIA developed the *DECIMER Image Classifier* and implemented the hand-drawn-like augmentation features. KR and HOB initiated, designed, tested, applied and validated the software features. CS and AZ conceived the project and supervised the work. All authors contributed to and approved the manuscript.

Competing interests

AZ is co-founder of GNWI - Gesellschaft für naturwissenschaftliche Informatik mbH, Dortmund, Germany.

Funding

MIA was supported by the Ministry of Education, Youth and Sports of the Czech Republic under project LM2023052. HOB and CS are funded by the Carl-Zeiss-Foundation, and KR is funded by the German Research Foundation within the framework CRC1127 ChemBioSys.

References

1. Brinkhaus, H. O., Rajan, K., Schaub, J., Zielesny, A. & Steinbeck, C. Open data and algorithms for open science in AI-driven molecular informatics. *ChemRxiv* (2023).
2. Herres-Pawlis, S., Liermann, J. C. & Koepler, O. Research data in chemistry – results of the first NFDI4Chem community survey. *Z. Anorg. Allg. Chem.* **646**, 1748–1757 (2020).
3. Steinbeck, C. *et al.* NFDI4Chem - Towards a National Research Data Infrastructure for Chemistry in Germany. *Riogrande Odontol.* **6**, e55852 (2020).
4. NFDI4Chem. nmrXiv - Open, FAIR and Consensus-Driven NMR spectroscopy data repository and analysis platform. *nmrXiv - Open, FAIR and Consensus-Driven NMR spectroscopy data repository and analysis platform* https://nmrxiv.org/ (2022).
5. Kearnes, S. M. *et al.* The Open Reaction Database. *J. Am. Chem. Soc.* **143**, 18820–18826 (2021).
6. Kim, S. *et al.* PubChem Protein, Gene, Pathway, and Taxonomy Data Collections: Bridging Biology and Chemistry through Target-Centric Views of PubChem Data. *J. Mol. Biol.* **434**, 167514 (2022).
7. wwPDB consortium. Protein Data Bank: the single global archive for 3D macromolecular structure data. *Nucleic Acids Res.* **47**, D520–D528 (2019).
8. Swain, M. C. & Cole, J. M. ChemDataExtractor: A Toolkit for Automated Extraction of Chemical Information from the Scientific Literature. *J. Chem. Inf. Model.* **56**, 1894–1904

(2016).

9. Contreras, M. L., Leonor Contreras, M., Allendes, C., Tomas Alvarez, L. & Rozas, R. Computational perception and recognition of digitized molecular structures. *Journal of Chemical Information and Modeling* **30,** 302–307 (1990).

10. Rozas, R. & Fernandez, H. Automatic processing of graphics for image databases in science. *J. Chem. Inf. Comput. Sci.* **30**, 7–12 (1990).

11. McDaniel, J. R. & Balmuth, J. R. Kekule: OCR-optical chemical (structure) recognition. *J. Chem. Inf. Comput. Sci.* **32**, 373–378 (1992).

12. Filippov, I. V. & Nicklaus, M. C. Optical structure recognition software to recover chemical information: OSRA, an open source solution. *J. Chem. Inf. Model.* **49**, 740–743 (2009).

13. Smolov, V., Zentsev, F. & Rybalkin, M. Imago: Open-Source Toolkit for 2D Chemical Structure Image Recognition, *Proceedings of Text Retrieval Conference* (2011).

14. Peryea, T., Katzel, D., Zhao, T., Southall, N. & Nguyen, D.-T. MOLVEC: Open source library for chemical structure recognition. *Abstracts of Papers of the American Chemical Society* **258,** (2019).

15. Clevert, D.-A., Le, T., Winter, R. & Montanari, F. Img2Mol - Accurate SMILES Recognition from Molecular Graphical Depictions. *Chem. Sci.* **12**, 14174-14181 (2021).

16. Staker, J., Marshall, K., Abel, R. & McQuaw, C. M. Molecular Structure Extraction from Documents Using Deep Learning. *J. Chem. Inf. Model.* **59**, 1017–1029 (2019).

17. Rajan, K., Zielesny, A. & Steinbeck, C. DECIMER: towards deep learning for chemical image recognition. *J. Cheminform.* **12**, 65 (2020).

18. Rajan, K., Zielesny, A. & Steinbeck, C. DECIMER 1.0: deep learning for chemical image recognition using transformers. *J. Cheminform.* **13**, 61 (2021).

19. Rajan, K., Brinkhaus, H. O., Zielesny, A. & Steinbeck, C. A review of optical chemical structure recognition tools. *J. Cheminform.* **12**, 60 (2020).

20. Musazade, F., Jamalova, N. & Hasanov, J. Review of techniques and models used in

optical chemical structure recognition in images and scanned documents. *J. Cheminform.* **14**, 61 (2022).

21. Oldenhof, M., Arany, A., Moreau, Y. & Simm, J. ChemGrapher: Optical Graph Recognition of Chemical Compounds by Deep Learning. *J. Chem. Inf. Model.* **60**, 4506–4517 (2020).
22. Khokhlov, I., Krasnov, L., Fedorov, M. V. & Sosnin, S. Image2SMILES: Transformer‐based molecular optical recognition engine. *Chemistry Methods* **2**, (2022).
23. Xu, Y. *et al.* MolMiner: You Only Look Once for Chemical Structure Recognition. *J. Chem. Inf. Model.* **62**, 5321-5328 (2022).
24. Xu, Z., Li, J., Yang, Z., Li, S. & Li, H. SwinOCSR: end-to-end optical chemical structure recognition using a Swin Transformer. *J. Cheminform.* **14**, 41 (2022).
25. He, K., Gkioxari, G., Dollár, P. & Girshick, R. Mask R-CNN. *arXiv* **[cs.CV]**, (2017).
26. Rajan, K., Brinkhaus, H. O., Sorokina, M., Zielesny, A. & Steinbeck, C. DECIMER-Segmentation: Automated extraction of chemical structure depictions from scientific literature. *J. Cheminform.* **13**, 20 (2021).
27. DECIMER Web Application. https://decimer.ai.
28. Willighagen, E. L. *et al.* The Chemistry Development Kit (CDK) v2.0: atom typing, depiction, molecular formulas, and substructure searching. *J. Cheminform.* **9**, 33 (2017).
29. Landrum, G. & Others. RDKit: Open-Source Cheminformatics Software.(2016). *URL http://www. rdkit. org/, https://github. com/rdkit/rdkit* (2016).
30. Indigo Toolkit. https://lifescience.opensource.epam.com/indigo/.
31. Terlouw, B. R., Vromans, S. P. J. M. & Medema, M. H. PIKAChU: a Python-based informatics kit for analysing chemical units. *J. Cheminform.* **14**, 34 (2022).
32. Tanimoto, T. T. Elementary mathematical theory of classification and prediction. (1958).
33. Jaccard, P. The distribution of the flora in the alpine zone.1. *New Phytol.* **11**, 37–50 (1912).
34. Papineni, K., Roukos, S., Ward, T. & Zhu, W.-J. Bleu: a method for automatic evaluation of machine translation. in *Proceedings of the 40th annual meeting of the Association for*

Computational Linguistics 311–318 (2002).

35. Qian, Y., Tu, Z., Guo, J., Coley, C. W. & Barzilay, R. Robust Molecular Image Recognition: A Graph Generation Approach. *arXiv* **[cs.CV]** (2022).

36. Brinkhaus, H. O., Zielesny, A., Steinbeck, C. & Rajan, K. DECIMER-hand-drawn molecule images dataset. *J. Cheminform.* **14**, 36 (2022).

37. Brinkhaus, H. O., Rajan, K., Zielesny, A. & Steinbeck, C. RanDepict: Random chemical structure depiction generator. *J. Cheminform.* **14**, 31 (2022).

38. Karulin, B. & Kozhevnikov, M. Ketcher: web-based chemical structure editor. *J. Cheminform.* **3**, 1 (2011).

39. Valko, A. T. & Johnson, A. P. CLiDE Pro: the latest generation of CLiDE, a tool for optical chemical structure recognition. *J. Chem. Inf. Model.* **49**, 780–787 (2009).

40. Mavračić, J., Court, C. J., Isazawa, T., Elliott, S. R. & Cole, J. M. ChemDataExtractor 2.0: Autopopulated Ontologies for Materials Science. *J. Chem. Inf. Model.* **61**, 4280–4289 (2021).

41. Isazawa, T. & Cole, J. M. Single Model for Organic and Inorganic Chemical Named Entity Recognition in ChemDataExtractor. *J. Chem. Inf. Model.* **62**, 1207–1213 (2022).

42. Beard, E. J., Sivaraman, G., Vázquez-Mayagoitia, Á., Vishwanath, V. & Cole, J. M. Comparative dataset of experimental and computational attributes of UV/vis absorption spectra. *Sci Data* **6**, 307 (2019).

43. Court, C. J. & Cole, J. M. Auto-generated materials database of Curie and Néel temperatures via semi-supervised relationship extraction. *Sci Data* **5**, 180111 (2018).

44. Beard, E. J. & Cole, J. M. Perovskite- and Dye-Sensitized Solar-Cell Device Databases Auto-generated Using ChemDataExtractor. *Sci Data* **9**, 329 (2022).

45. Huang, S. & Cole, J. M. A database of battery materials auto-generated using ChemDataExtractor. *Sci Data* **7**, 260 (2020).

46. Decimer-segmentation. *PyPI* https://pypi.org/project/decimer-segmentation/.

47. Rajan, K., Brinkhaus, H. O., Sorokina, M., Zielesny, A. & Steinbeck, C. *DECIMER-Image-Segmentation - GitHub*. (2022). https://github.com/Kohulan/DECIMER-Image-Segmentation

48. Rajan, K., Brinkhaus, H. O., Zielesny, A. & Steinbeck, C. DECIMER-Segmentation model. doi.org/10.5281/ZENODO.7228583 (2021).

49. Kim, S. *et al.* PubChem in 2021: new data content and improved web interfaces. *Nucleic Acids Res.* **49**, D1388–D1395 (2021).

50. Ashton, M. *et al.* Identification of diverse database subsets using property-based and fragment-based molecular descriptions. *Quant. struct.-act. relatsh.* **21**, 598–604 (2002).

51. Dalke, A. The chemfp project. *J. Cheminform.* **11**, 76 (2019).

52. O'Boyle, N. & Dalke, A. DeepSMILES: An Adaptation of SMILES for Use in Machine-Learning of Chemical Structures. *ChemRxiv* (2018).

53. Krenn, M., Häse, F., Nigam, A., Friederich, P. & Aspuru-Guzik, A. Self-referencing embedded strings (SELFIES): A 100% robust molecular string representation. *Mach. Learn.: Sci. Technol.* **1**, 045024 (2020).

54. Rajan, K., Steinbeck, C. & Zielesny, A. Performance of chemical structure string representations for chemical image recognition using transformers. *Digital Discovery* **1**, 84–90 (2022).

55. Chollet, F. & Others. Keras. https://keras.io (2015).

56. Abadi, M. *et al.* TensorFlow: Large-Scale Machine Learning on Heterogeneous Distributed Systems. *arXiv* **[cs.DC]** (2016).

57. Weir, H. *et al.* ChemPix: automated recognition of hand-drawn hydrocarbon structures using deep learning. *Chem. Sci.* **12**, 10622–10633 (2021).

58. Vaswani, A. et al. Attention Is All You Need. arXiv **[cs.CL]** (2017) .

59. Tan, M. & Le, Q. V. EfficientNetV2: Smaller Models and Faster Training. *arXiv* **[cs.CV]** (2021).

60. Rajan, K. *OCSR_Review: This repository contains the information related to the benchmark study on openly available OCSR tools*. (Github). https://github.com/Kohulan/OCSR_Review

61. OSRA validation datasets. https://sourceforge.net/p/osra/wiki/Validation/.

62. Sadawi, N. M., Sexton, A. P. & Sorge, V. Chemical structure recognition: a rule-based approach. in *Document Recognition and Retrieval XIX* **8297**, 101–109 (2012).

63. CLEF dataset. https://www.ifs.tuwien.ac.at/~clef-ip/download/2012/qrels/clef-ip-2012-chem-recognition-qrels.tgz.

64. *docker-osra: OSRA (Optical Structure Recognition Application) in Docker*. (Github). https://github.com/daverona/docker-osra

65. Docker. https://hub.docker.com/repository/docker/obrink/osra.

66. Download molvec JAR 0.9.8 ➔ With all dependencies! https://jar-download.com/artifacts/gov.nih.ncats/molvec/0.9.8/source-code.

67. epam. *Imago*. (2013).

68. Brinkhaus, O. *Img2Mol_standalone at f8143858cac1aabad348fe79448abf5328a853fc*. (Github). https://github.com/OBrink/Img2Mol_standalone/tree/f8143858cac1aabad348fe79448abf5328a853fc

69. *SwinOCSR*. (Github). https://github.com/suanfaxiaohuo/SwinOCSR

70. Gaulton, A. *et al.* ChEMBL: a large-scale bioactivity database for drug discovery. *Nucleic Acids Res.* **40**, D1100–7 (2012).

71. Sorokina, M., Merseburger, P., Rajan, K., Yirik, M. A. & Steinbeck, C. COCONUT online: Collection of Open Natural Products database. *J. Cheminform.* **13**, 2 (2021).

72. Youden, W. J. Index for rating diagnostic tests. *Cancer* **3**, 32–35 (1950).

73. Fluss, R., Faraggi, D. & Reiser, B. Estimation of the Youden Index and its associated cutoff point. *Biom. J.* **47**, 458–472 (2005).

74. Hastings, J. *et al.* ChEBI in 2016: Improved services and an expanding collection of metabolites. *Nucleic Acids Res.* **44**, D1214–9 (2016).

75. Zhong, X., Tang, J. & Jimeno Yepes, A. PubLayNet: Largest Dataset Ever for Document Layout Analysis. in *2019 International Conference on Document Analysis and Recognition ICDAR.* 1015–1022 (2019).

76. Rajan, K., Zielesny, A. & Steinbeck, C. STOUT: SMILES to IUPAC names using neural machine translation. *J. Cheminform.* **13**, 34 (2021).

Supplementary Information

Training and test of the *DECIMER Image Transformer* 299 x 299 model

For the models trained using datasets containing images with a size of 299 x 299 pixels, EfficientNet-V2-B3 was used without any further modifications. Training the model with the image size of 299 x 299 pixels was done using the TPU v3-32 pod slices. The per-node batch size was set to 128 for the models trained on datasets with images of 299 x 299 pixels. Training scripts and models are written in Python 3 with Keras and Tensorflow 2.8.0.

To test each model trained with the *pubchem_1* and *pubchem_2* datasets with an image size of 299 x 299 pixels, a pre-selected set of 250,000 molecules was used. Each of these molecules was depicted twice with an image size of 299 x 299 pixels, once without augmentations and once with augmentations. The model trained using *pubchem_2* was further evaluated by a set of 250,000 Markush structures, depicted with and without augmentations. The purpose of this was to evaluate the model's accuracy in detecting chemical structures depicted with R-Group representations.

Supplementary Table 1. Training datasets for *DECIMER Image Transformer*

Dataset ID	**Number of molecules**	**Number of depictions**	**Composition**
pubchem_1	107.5 Mio	430 Mio	1 clean and 3 augmented depictions per molecule with a size of 299 x 299 pixels
pubchem_2	127.5 Mio	479.5 Mio	All molecules from *pubchem_1* + 20 Mio depictions of Markush structures with a resolution of 299 x 299 pixels (1 clean and 3 augmented depictions per molecule)
pubchem_3	127.5 Mio	453.9 Mio	All molecules and Markush structures from *pubchem_2* with a resolution of 512 x 512 pixels (1 clean and 3 augmented depictions per molecule)
hand_drawn	127.5 Mio	127.5 Mio	1 depiction with augmentations with a resolution of 512 x 512 pixels that makes it appear like hand-drawn per molecule.

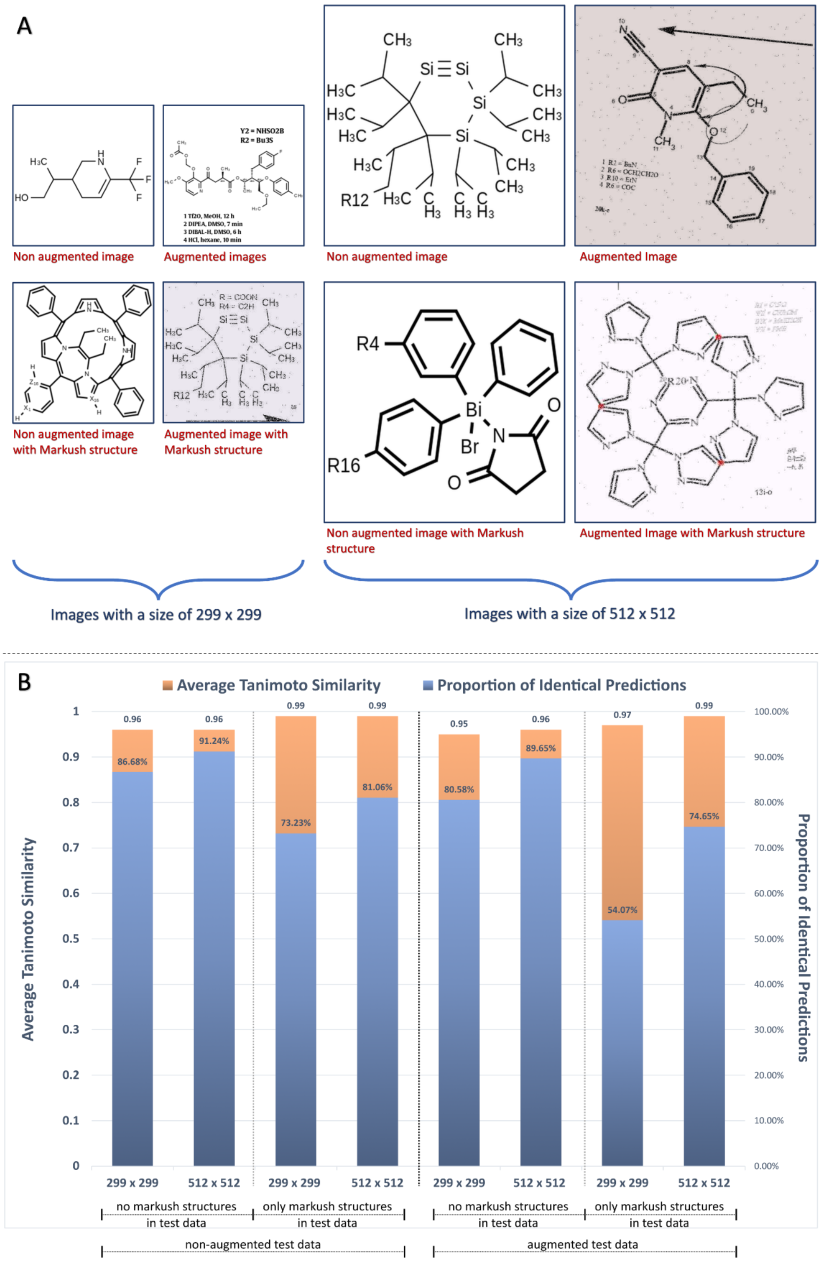
Supplementary Table 2. Test results analysis and model performance. Columns A, B and C contain the test results on images with a resolution of 299 x 299 pixels. Columns D and E contain the test results on images with a resolution of 512 x 512 pixels. Finally, columns F and G contain the BLEU scores for the test results from C and E.

	A		**B**		**C**	
Test data	299 x 299 depictions (no Markush structures)		299 x 299 depictions (no Markush structures)		299 x 299 depictions (with Markush structures)	
Model trained on	*pubchem_1* (no Markush structures)		*pubchem_2* (with Markush structures)		*pubchem_2* (with Markush structures)	
	Non-Augmented Images	Augmented Images	Non-Augmented Images	Augmented Images	Non-Augmented Images	Augmented Images
Valid Predictions	99.79%	99.56%	96.50%	96.60%	99.80%	99.79%
Identical Predictions	90.85%	85.62%	86.68%	80.58%	73.23%	54.07%
Tanimoto 1.0 Count	95.66%	90.68%	92.89%	88.21%	91.89%	83.07%
Average Tanimoto	0.99	0.98	0.96	0.95	0.99	0.97

	D		**E**	
Test data	512 x 512 depictions (without Markush structures)		512 x 512 depictions (with Markush structures)	
Model trained on	*pubchem_3*		*pubchem_3*	
	Non-Augmented Images	Augmented Images	Non-Augmented Images	Augmented Images
Valid Predictions	96.46%	96.54%	99.83%	99.81%
Identical Predictions	91.24%	89.65%	81.06%	74.65%
Tanimoto 1.0 Count	94.77%	93.47%	94.42%	92.06%
Average Tanimoto	0.96	0.96	0.99	0.99

BLEU Scores for test results with R-Group representations.

	F		G	
Test data	299 x 299 depictions (with Markush structures)		512 x 512 depictions (with Markush structures)	
Model trained on	*pubchem_2*		*pubchem_3*	
BLEU Scores	Non-Augmented Images	Augmented Images	Non-Augmented Images	Augmented Images
BLEU-1:	0.96	0.95	0.97	0.97
BLEU-2:	0.96	0.94	0.97	0.96
BLEU-3:	0.95	0.93	0.96	0.96
BLEU-4:	0.94	0.91	0.96	0.95
Average	0.94	0.91	0.96	0.95



Supplementary Figure 1: Representation of types of images in the training and the test datasets. **B:** In-domain test results of two models trained and tested using images with a resolution of 299 x 299 or 512 x 512 pixels, respectively. All training datasets include depictions of Markush structures and a variety of image augmentations (datasets *pubchem_2* and *pubchem_3* in Supplementary Table 1). In the test datasets, these features were separately evaluated as described in the text to assess their influence on performance. All in-domain test results are also presented in Supplementary Table 2.

Supplementary Table 3. Benchmark results - performance of each model/tool on each dataset. The performance is described as the proportion of occurrences of identical predictions P_i and the average Tanimoto similarity $\overline{T}$. Section A. Benchmark results for datasets without added distortions. Section B. Benchmark results for datasets with added distortions, such as mild shearing and rotation.

A. Benchmark results for datasets without added distortions.

	JPO		CLEF		USPTO		UOB		USPTO Big		Indigo		Img2Mol Test		DECIMER-Hand drawn		DECIMER - Test non augmented	
	P_i	$\overline{T}$	P_i	$\overline{T}$	P_i	$\overline{T}$	P_i	$\overline{T}$	P_i	$\overline{T}$	P_i	$\overline{T}$	P_i	$\overline{T}$	P_i	$\overline{T}$	P_i	$\overline{T}$
OSRA	56%	0.78	**85%**	0.88	**88%**	0.96	78%	0.95	0.01%	0.17	2%	0.29	2%	0.14	1%	0.17	8%	0.33
MolVec	**66%**	0.89	83%	0.89	**88%**	0.97	80%	0.96	1%	0.35	2%	0.27	2%	0.29	1%	0.23	5%	0.33
Imago	40%	0.68	59%	0.85	87%	0.96	58%	0.87	0%	0.10	0.04%	0.08	0.02%	0.11	3%	0.22	2%	0.19
Img2Mol	15%	0.70	16%	0.81	24%	0.85	68%	0.94	16%	0.78	22%	0.59	**85%**	**0.97**	5%	0.52	16%	0.78
SwinOCSR	13%	0.75	29%	0.81	27%	0.88	45%	0.97	0.23%	0.68	0.20%	0.48	4%	0.53	5%	0.64	6%	0.54
MolScribe	50%	0.93	75%	0.89	79%	**0.99**	87%	0.99	**79%**	0.95	38%	0.65	51%	0.93	8%	0.59	44%	0.85
DECIMER 299	47%	0.89	55%	0.87	41%	0.93	87%	0.99	50%	0.92	54%	0.66	28%	0.85	38%	0.78	**91%**	**0.99**
DECIMER 512	64%	**0.93**	72%	**0.96**	61%	0.97	**88%**	**0.98**	63%	**0.97**	**60%**	**0.98**	55%	0.93	27%	0.69	91%	**0.99**
DECIMER 512 Fine Tuned	-	-	-	-	-	-	-	-	-	-	-	-	-	-	**60%**	**0.89**	-	-

B. Benchmark results for datasets with distortions

	JPO (dist)		CLEF (dist)		USPTO (dist)		UOB (dist)		USPTO_big (dist)		Indigo (dist)		DECIMER-Test augmented	
	P_i	$\overline{T}$	**P_i**	$\overline{T}$	**P_i**	$\overline{T}$	**P_i**	$\overline{T}$	**P_i**	$\overline{T}$	**P_i**	$\overline{T}$	**P_i**	$\overline{T}$
OSRA	38%	0.70	19%	0.66	7%	0.60	61%	0.90	0.01%	0.13	0.42%	0.16	2%	0.15
MolVec	41%	0.80	21%	0.66	26%	0.71	63%	0.92	0.02%	0.14	0.48%	0.07	1%	0.12
Imago	23%	0.47	33%	0.65	51%	0.81	34%	0.64	0%	0.08	0.01%	0.20	0.15%	0.10
Img2Mol	15%	0.67	15%	0.80	21%	0.83	70%	0.94	1%	0.56	15%	0.54	1%	0.60

SwinOCSR	7%	0.71	21%	0.81	23%	0.87	6%	0.95	0%	0.38	0.01%	0.38	0.18%	0.36
MolScribe	52%	**0.93**	**73%**	0.89	**75%**	**0.99**	86%	**0.99**	**78%**	0.95	34%	0.64	9%	0.53
DECIMER 299	52%	0.91	58%	0.87	45%	0.94	**86%**	**0.99**	28%	0.83	34%	0.62	**86%**	**0.98**
DECIMER 512	**62%**	**0.93**	72%	**0.96**	61%	0.96	**86%**	0.98	57%	**0.96**	**51%**	**0.97**	**90%**	**0.99**

Supplementary Table 4. Catastrophic and severe failure rates of each model/tool on each dataset. **T_E**: Percentage of predictions with Tanimoto similarity values of zero and invalid predictions (catastrophic failure). **$T_{<=0.3}$**: The percentage of predictions with Tanimoto similarity less than or equal to 0.3 (severe failure) . **A:** Benchmark results for datasets without added distortions. **B:** Benchmark results for datasets with added distortions, such as mild shearing and rotation.

A. Benchmark results for datasets without added distortions.																		
	JPO		CLEF		USPTO		UOB		USPTO Big		Indigo		Img2Mol Test		DECIMER-Hand drawn		DECIMER - Test non augmented	
	T_E	$T_{<=0.3}$	T_E	$T_{<=0.3}$	T_E	$T_{<=0.3}$	T_E	$T_{<=0.3}$	T_E	$T_{<=0.3}$	T_E	$T_{<=0.3}$	T_E	$T_{<=0.3}$	T_E	$T_{<=0.3}$	T_E	$T_{<=0.3}$
OSRA	14%	19%	4%	4%	2%	2%	2%	2%	8%	92%	25%	42%	34%	63%	49%	80%	43%	58%
MolVec	6%	8%	3%	3%	2%	2%	2%	2%	21%	45%	23%	29%	28%	91%	57%	68%	32%	41%
Imago	23%	26%	7%	7%	3%	3%	6%	7%	19%	98%	25%	96%	36%	54%	34%	73%	35%	79%
Img2Mol	2%	7%	3%	3%	3%	3%	1%	1%	1%	2%	1%	2%	**0%**	**0%**	**2%**	28%	4%	4%
SwinOCSR	6%	9%	5%	6%	2%	3%	0.21%	0.33%	3%	6%	5%	8%	8%	12%	3%	**13%**	11%	28%
MolScribe	**1%**	**2%**	3%	3%	**0.38%**	**0.5%**	0.02%	0.02%	**0.22%**	**0.29%**	1%	1%	1%	2%	5%	19%	**2%**	**3%**
DECIMER 512	3%	4%	**2%**	**2%**	1%	1%	**0%**	**0%**	0.26%	0.47%	**0.20%**	**0.21%**	2%	3%	5%	18%	4%	4%

B. Benchmark results for datasets with distortions

	JPO (dist)		CLEF (dist)		USPTO (dist)		UOB (dist)		USPTO_big (dist)		Indigo (dist)		DECIMER-Test augmented	
	T_E	$T_{<=0.3}$	T_E	$T_{<=0.3}$	T_E	$T_{<=0.3}$	T_E	$T_{<=0.3}$	T_E	$T_{<=0.3}$	T_E	$T_{<=0.3}$	T_E	$T_{<=0.3}$
OSRA	18%	23%	19%	20%	25%	26%	4%	5%	11%	97%	25%	62%	62%	81%
MolVec	11%	12%	12%	13%	15%	16%	27%	29%	4%	77%	20%	33%	33%	41%
Imago	42%	46%	28%	29%	16%	16%	3%	3%	9%	100%	28%	105%	42%	92%
Img2Mol	3%	7%	3%	4%	3%	3%	1%	1%	1%	6%	1%	3%	4%	8%
SwinOCSR	5%	11%	5%	6%	2%	3%	0.14%	0.23%	7%	29%	7%	17%	29%	47%
MolScribe	**0.44%**	**1%**	3%	3%	**0.35%**	**0.38%**	**0%**	**0%**	**0.23%**	**0.30%**	1%	1%	19%	29%
DECIMER 512	3%	4%	**2%**	**2%**	1%	1%	**0%**	**0%**	0.39%	1%	**0.16%**	**0.19%**	**3%**	**3%**

Supplementary Table 5 : Test performance of a model trained on SMILES strings without further modifications (Model 1) and SMILES strings with replaced R-group indices (Model 2)

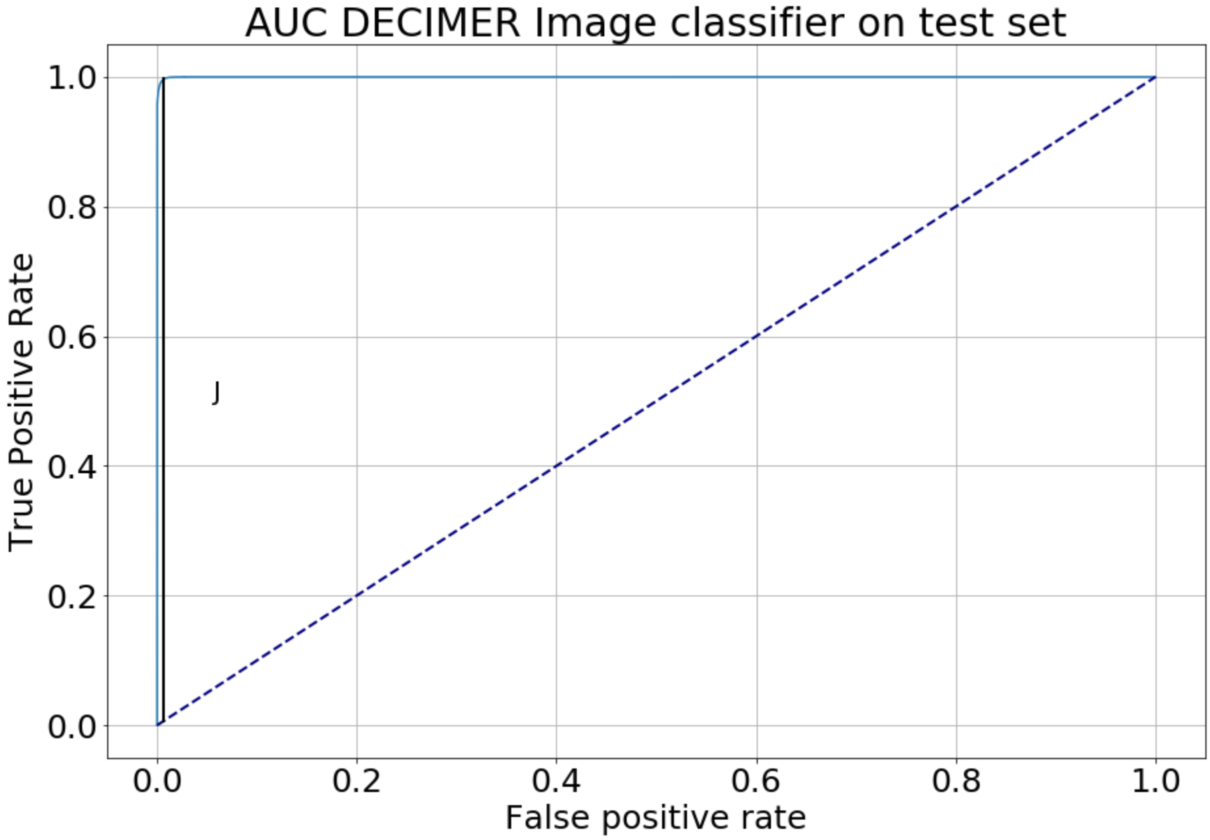
	Model 1 (SMILES)	**Model 2 (modified SMILES)**
Valid Predictions	95.70%	99.33%
Identical Predictions	77.87%	77.37%
Tanimoto 1.0 Count	86.55%	88.78%
Average Tanimoto similarity	0.94	0.98

Supplementary Table 6. Datasets used as non-chemical structures to train, validate and test the *DECIMER Image Classifier*.

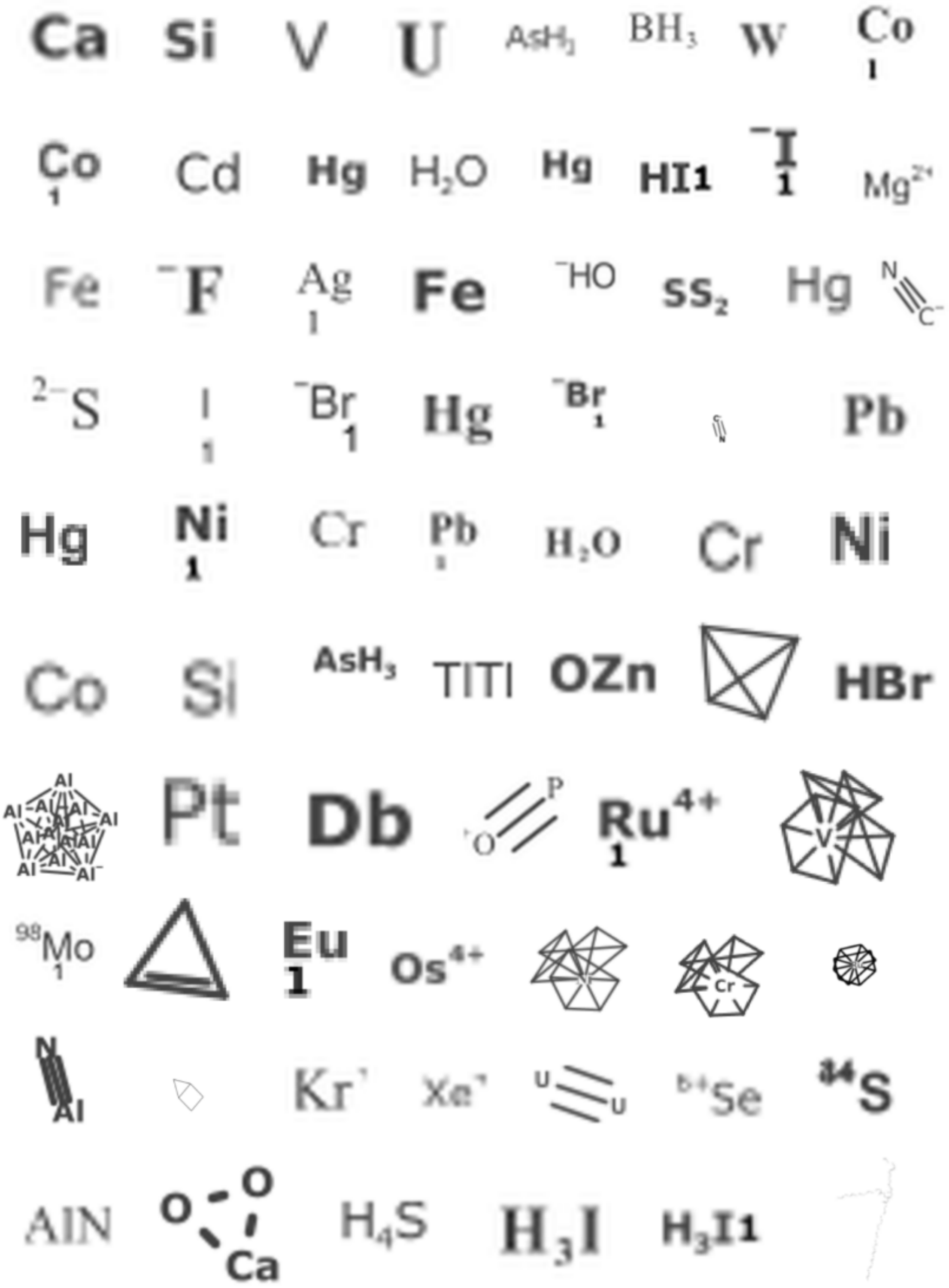
Dataset name	**Number of images**	**Modifications**	**Reference**
Places-205	2462123	None	B. Zhou, A. Lapedriza, J. Xiao, A. Torralba, and A. Oliva. Learning Deep Features for Scene Recognition

			using Places Database. Advances in Neural Information Processing Systems 27 (NIPS), 2014.
COCO	287360	None	Lin, Tsung-Yi et al. (2014). Microsoft COCO: Common Objects in Context. https://arxiv.org/abs/1405.0312
Google Open labelled Images	1909039	None	https://storage.googleapis.com/openimages/web/index.html
MMU-OCR-21	301229	None	T. Nasir, M. K. Malik and K. Shahzad, "MMU-OCR-21: Towards End-to-End Urdu Text Recognition Using Deep Learning," in IEEE Access, doi: 10.1109/ACCESS.2021.3110787
HandWritten_Character	821715	None	https://www.kaggle.com/datasets/vaibhao/handwritten-characters
CoronaHack -Chest X-Ray-	5933	None	https://www.kaggle.com/datasets/praveengovi/coronahack-chest-xraydataset
PANDAS Augmented Images	12083	None	https://www.kaggle.com/datasets/amyjang/pandatilesagg?select=all_images
Bacterial_Colony	681	None	https://www.kaggle.com/datasets/nilay1987/bacterial-colony
Ceylon Epigraphy Periods	5149	Colour inversion	https://www.kaggle.com/datasets/pabasar/ceylon-epigraphy-periods
Chinese Calligraphy Styles by Calligraphers	105129	None	https://www.kaggle.com/datasets/yuanhaowa

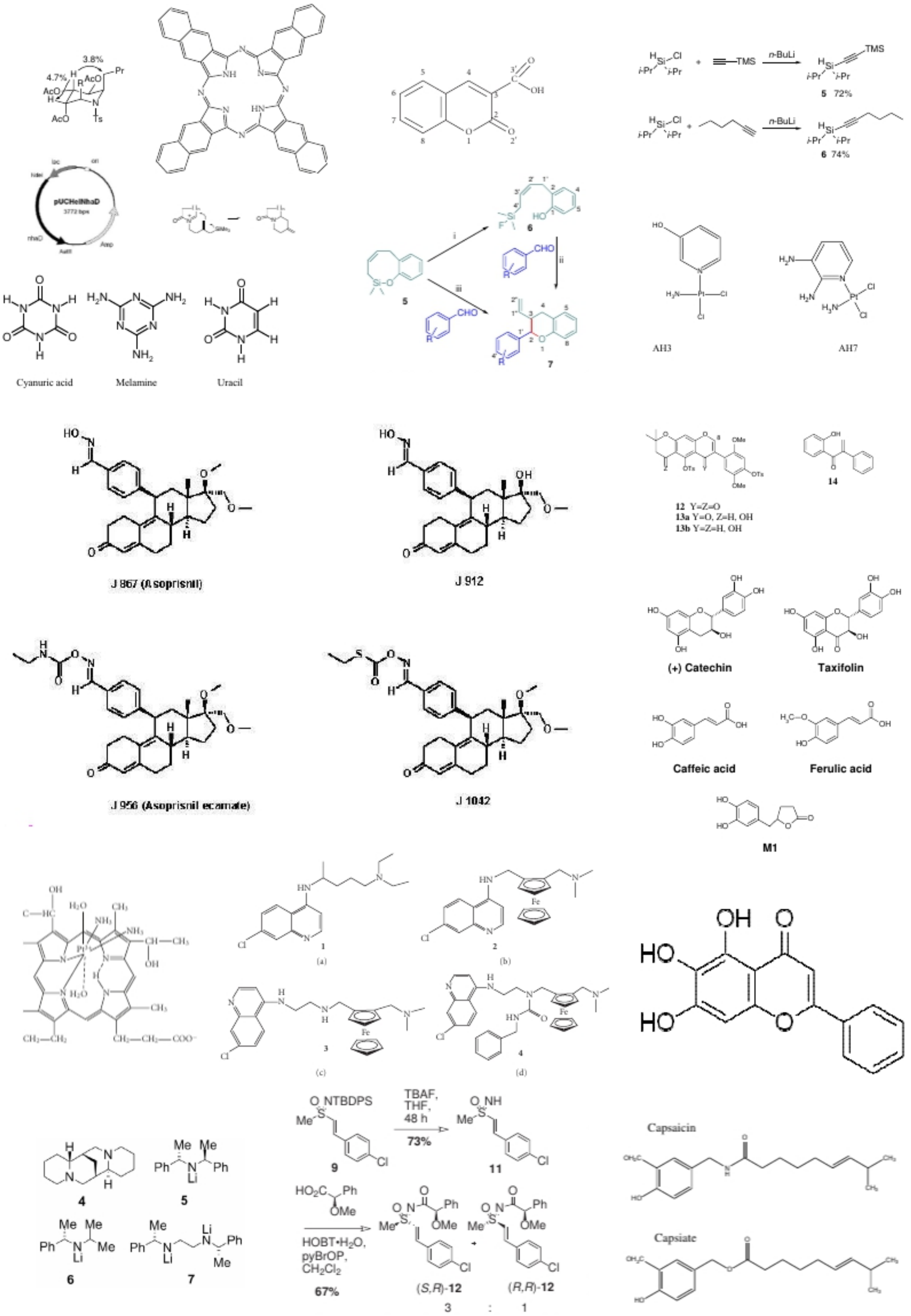
			ng486/chinese-calligraphy-styles-by-calligraphers
Graphs Dataset	15861	None	https://www.kaggle.com/datasets/sunedition/graphs-dataset
Function_Graphs Polynomial	10250	None	https://www.kaggle.com/datasets/kopfgeldjaeger/function-graphs-polynomial
sketches	20000	None	https://www.kaggle.com/datasets/vishnunkumar/sketches
Person Face Sketches	16909	None	https://www.kaggle.com/datasets/almightyj/person-face-sketches
Art Pictograms	3545	None	https://www.kaggle.com/datasets/olgabelitskaya/art-pictogram
Russian handwritten letters	219183	Monochrome and 50% colour conversion	https://www.kaggle.com/datasets/olgabelitskaya/handwritten-russian-letters https://www.kaggle.com/datasets/tatianasnwrt/russian-handwritten-letters
Covid-19 Misinformation Tweets Labelled	13304	Monochrome, remap and 50% colour conversion	https://www.kaggle.com/datasets/arashnic/misinfo-graph
grapheme-imgs-224x224	200840	Colour inversion	https://www.kaggle.com/datasets/roycezjq/graphemeimgs224x224



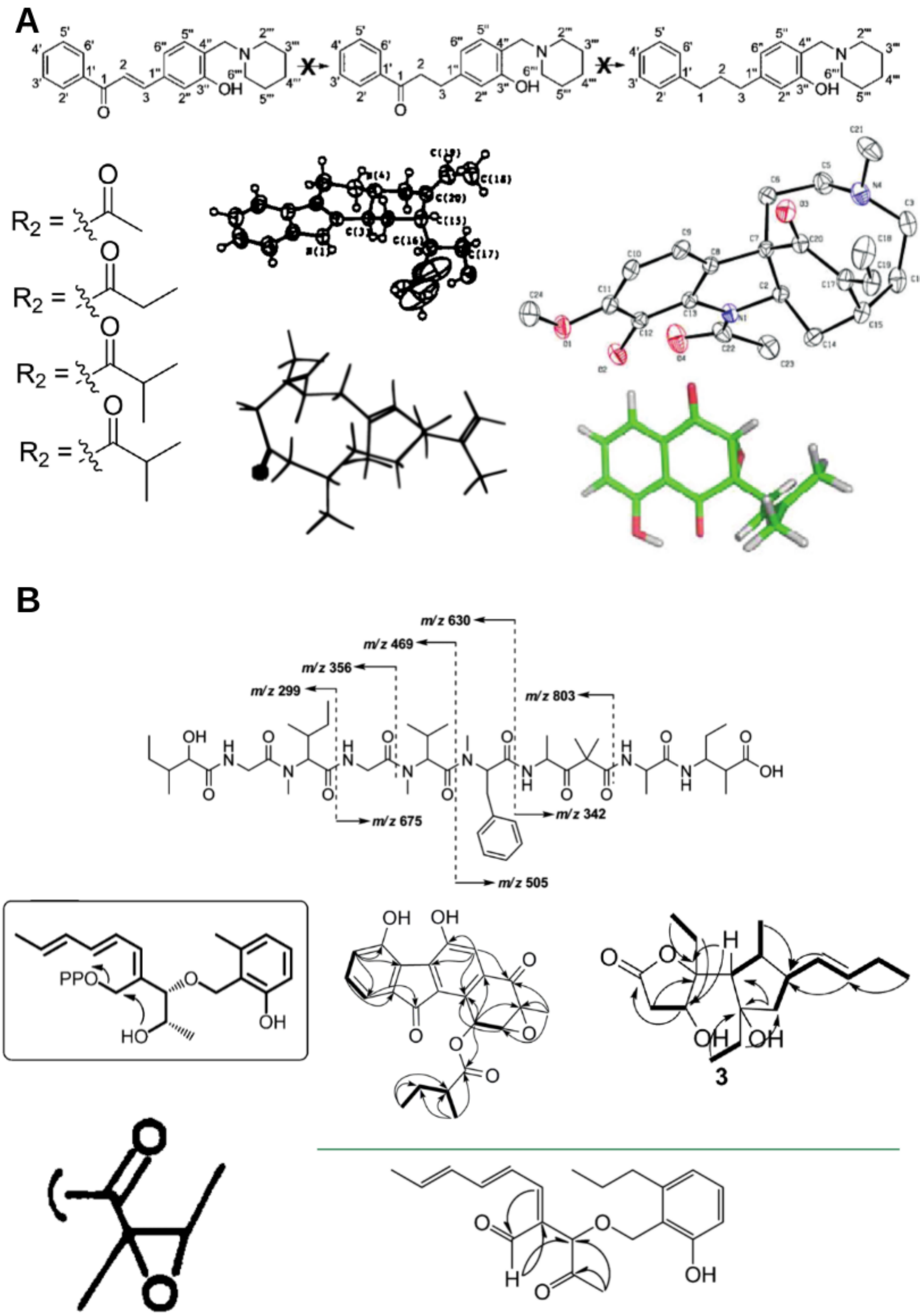
Supplementary Figure 2. AUC of the *DECIMER Image classifier* predictions on the test set. Dotted dark blue marks the random chance, solid light blue represents the ROC curve and the black vertical line represents the Youden index.



Supplementary Figure 3. Examples of non-chemical structures predicted for the *ChEBI* dataset.



Supplementary Figure 4. Examples of chemical structures predicted on the test dataset *PubLayNet*.



Supplementary Figure 5. Examples of wrong JNP segment classifications by the *DECIMER-Image classifier*. A) False positives. B) False Negative.

More detailed presentation of the *DECIMER Image Classifier* out-of-domain test results

Concerning the three public out-of-domain datasets, we used one that only consists of chemical structures (*ChEBI*), one without any chemical structures (*EM_Images*) and one that mostly contains no chemical structures (*PubLayNet*). The performance of the *DECIMER Image Classifier* on these datasets is as follows,

- *ChEBI*: 97.33% of the images were correctly classified as chemical structure depictions.
- *EM_Images*: 100% of the images were correctly classified as images without chemical structures.
- *PubLayNet*: 99.13% of the images were correctly classified as images without chemical structure depictions.

Furthermore, the performance of the *DECIMER Image Classifier* was evaluated using 8,733 images segmented from 1,000 publications from the Journal of Natural Products (JNP). Based on a manual inspection of the results, 8,187 of the 8,733 images were true positives, 178 were true negatives, 47 were false positives, and 321 were false negatives. As a result of computing the same performance metrics used for the test set, the results were AUC = 0.94, MCC = 0.51, accuracy = 0.96, specificity = 0.79, and sensitivity = 0.96.

It should be noted, however, that the calculated MCC was slightly lower than that calculated on the test set. It is primarily the characteristics used to determine whether an image depicts a chemical structure or not. Most of the false positive classifications are 3D chemical depictions that can not be interpreted as normal chemical structure depictions. Most of the false negative classifications are due to the presence of arrows in the image. The wrong predictions produced by the *DECIMER Image Classifier* are illustrated in Supplementary Figure 3-5.

Code Resource 1: Content of the script img2mol_batch_run.py that was used to run the standalone version [68] of Img2Mol in our benchmark.

```python
import sys
import os
from img2mol.inference import *


def main():
    """
    This script takes three arguments:
    1) path of the directory with images to process with img2mol
    2) output directory
    3) file ending of images in the directory given at 1

    It runs img2mol on all images with the given file ending in the
    input directory and saves a file with the results in the output
    directory.
    """
    input_dir, output_dir, file_ending = sys.argv[1:]
    im_names = [img for img in os.listdir(input_dir)
                if img[-len(file_ending):].lower() == file_ending.lower()]
    img2mol = Img2MolInference(local_cddd=True)
    output_file_path = os.path.join(output_dir, "img2mol_results.smiles")
    with open(output_file_path, "w") as output_file:
        for im_name in im_names:
            im_path = os.path.join(input_dir, im_name)
            result = img2mol(filepath=im_path)
            smiles = result['smiles']
            output_file.write(f"{im_name}\t{smiles}\n")


if __name__ == "__main__":
    if len(sys.argv) == 4:
        main()
    else:
        print(f"Usage: {sys.argv[0]} input_dir output_dir file_ending")
```

3 Discussion

The extraction of chemical information from the scientific literature comprises the segmentation of chemical structure depictions and their subsequent translation into structured, machine-readable representations. The deep-learning-based applications DECIMER Segmentation (Publication A) and DECIMER Image Transformer (Publication E) represent solutions for these problems. Additionally, RanDepict (Publication B) is the implementation of a OCSR artificial training data generation and diversification strategy that has been successfully used during the development of DECIMER Image Transformer (Publication E). The DECIMER hand-drawn molecule image dataset (Publication C) is an open and FAIR resource for the OCSR research community. The developed methods for the segmentation, classification and recognition of chemical structure depictions have been integrated into the DECIMER.ai web application (Publication E) that enables the automated extraction of molecular structures from the literature in a user interface. The applications described herein have been developed with the aim of making chemical information publicly available in structured data formats. The recent progress that has been driven by the combination of artificially intelligent deep-learning methods and the open availability of chemical data has been assessed (Publication D). The work described in this thesis contributes to automated chemical information extraction with deep-learning methods.

3.1 DECIMER Segmentation

Until today, DECIMER Segmentation is the only deep learning-based open-source tool for segmenting chemical structure depictions from printed literature (see Publication A). It is the implementation of a robust solution for this fundamental problem in the chemical literature mining workflow.

Other tools offer segmentation functionalities but are not openly available, or their use cases are much more limited. OSRA has a basic in-built segmentation system that determines whether a region contains a chemical structure based on its proportion of black pixels and the size and shape of the region [95]. MolMiner has a deep learning-based image segmentation module based on Deeplab V3 [125, 129], but the source code and the model weights are not openly available. Staker et al. propose a segmentation method based on the fully convolutional U-Net architecture, but the model is not publicly available [94, 105]. The open-source tool ChemSchematicResolver is capable of recognising chemical structure depictions in images where they are presented in combination with text labels exclusively. This is done by applying k-means clustering (with $k = 2$) to all elements in the image based on a feature density metric [130]. ReactionDataExtractor recognises chemical structure depictions, arrows and different types of text labels in images with chemical reaction schemes and returns this information in a structured data format [131]. Here, the segmentation of the chemical structure depictions is based on the unsupervised machine learning method Density-Based Spatial Clustering of Application with Noise (DBSCAN) [132]. As for ChemSchematicResolver, the segmentation procedure is based on a clustering method that only works in the specific context of a chemical reaction scheme. However, the model does not encode a generally applicable concept of what a chemical structure looks like. Opposed to the other openly available approaches listed here, the Mask R-CNN model of DECIMER Segmentation follows a more complex approach to distinguish between chemical structures and other objects on a page. As it uses CNNs, it can learn to recognise chemical structures independent of the type of surrounding elements instead of just clustering objects based on feature density metrics in more narrowly defined use cases like ChemSchematicResolver and ReactionDataExtractor.

DECIMER Segmentation was only trained on a comparably small set of manually annotated pages from the Journal of Natural Products. Consequently, the tool performs well on document pages with a similar format to those in the training data. On the other hand, the tool tends to fail when it processes images that only contain chemical structure depictions that are not embedded in whole article pages. In the future, the application should be further developed to be able to recognise chemical structure depictions equally well in any image. One key aspect for implementing this type of capability in DECIMER Segmentation is modifying the training data. To this end, an artificial training data generation strategy could be developed as it has been done for the training data generation for DECIMER Image Transformer (Publication E) using RanDepict (Publication B). Here, artificially generated structure depictions could be inserted in different kinds of images. As the structures would be inserted automatically, the region where they have been inserted would be known and could be used as an annotation for the training of a segmentation model. This way, large amounts of training data could be generated in an automated manner so that a model trained on that data could learn to recognise chemical structure depictions in any environment. Another aspect of DECIMER Segmentation that could be improved in the future is the segmentation model that is running at its core.

3.2 DECIMER Image Transformer

DECIMER Image Transformer is an deep learning-based OCSR application capable of translating various chemical structure depictions into SMILES representations (see Publication E). Opposed to tools like ChemPix, which can only be used for hand-drawn structures that exclusively contain carbon and hydrogen atoms [120] or Img2Mol, which is incapable of interpreting chirality or Markush structures, DECIMER Image Transformer can be successfully applied to most types of chemical structure depictions in the literature (see Publication E). The comprehensive training data diversification strategy developed as a part of this thesis (see Publication B) contributed to DECIMER Image Transformer outperforming all available tools in a comparative performance evaluation on independent test datasets (see Publication E).

Currently, DECIMER Image Transformer focuses on small molecules with an atomic mass of less than 1500 Dalton (see Publication E). In the future, this could be expanded to depictions of macromolecules as well. Another limitation is the incapability to read random functional group, superatom and R-group labels. The training data includes a variety of common labels, but the model can only read those labels that it has learned to recognise during training. Inspired by what the developers of Molscribe implemented [126], the capability to read any text label as a superatom could be achieved by adding random text labels to the training data. Known labels could then be replaced based on a dictionary.

3.3 DECIMER.ai - An open platform for chemical literature mining

Integrating all DECIMER components in a comprehensive graphical user interface offers users without any programming knowledge a chemical literature extraction workflow. A document file or images with chemical structure depictions can be uploaded and processed automatically. The visualisation of the results in a molecular editor window [133] next to the segmented structure depictions offers the option of a straightforward assessment and potential correction of the results by a human curator. This way, the DECIMER.ai application (Publication E) has the potential to reduce the manual curation effort for the generation of chemical databases enormously.

There are two other existing systems that combine the segmentation of chemical structures and OCSR in a graphical user interface: CLiDE [98, 99] and Molminer [125]. CLiDE is a fully commercial tool, and MolMiner is freely available with a license that enables limited usage, but both are closed-source applications. Opposed to these systems, users can use, deploy or modify the DECIMER.ai application according to their specific needs as it has been published under a permissive licence.

Implementing text mining functionalities would be a sensible addition to the workflow of DECIMER.ai. The chemical text mining toolkit ChemDataExtractor [134] has been actively developed over the past years [135] and uses a chemical NER system based on BERT [136] that is capable of recognising inorganic and organic chemical names equally well [73]. Since its first release, ChemDataExtractor has been used widely for the automated generation of databases containing magnetic properties of semiconductors [137], properties of battery materials [74, 138], UV/Vis spectra[139], refractive indices and dielectric constants [140], semiconductor band gaps [141], dye-sensitised solar cell materials [142] and thermoelectric materials [143]. There are different approaches like MatSciBERT, a language model that has been trained on a large set of publications from the domain of materials science [75]. Integrating such systems into DECIMER.ai would complement the existing image-processing functionalities and lead to a chemical literature extraction system capable of dealing with a wider variety of published chemical knowledge.

The implementation of OCR for recognising labels associated with chemical structure depictions will be necessary to link the information from text and images. In the chemical literature, compounds are commonly referred to by labels like '*1a*'. These labels are the binding element between image and text contents in the literature as they are used in the text and the images. Sometimes, the labels also contain specifications for the R-group variables in the depictions. The above-mentioned tool ChemSchematicResolver can be used to link these types of labels to the associated chemical structures [130]. Still, its usage is strictly limited to images that only contain text labels and structure depictions.

This way, the DECIMER.ai application could develop into a chemical literature mining platform capable of extracting all types of chemical information from the scientific literature. Chemical structures could be segmented and translated into SMILES representations using DECIMER Segmentation and DECIMER Image Transformer. The ID and R-group labels could be detected and recognised, and linked to the chemical structures. Then, information could be extracted from the text contents using a chemical text mining application. As the ID labels are used in the images and the text, they can be used to link the resolved molecular structures to the information that has been extracted from the text, like chemical names and spectral data. Then the information could be returned in a machine-readable, structured data format.

3.4 DECIMER Hand-Drawn Molecule Images dataset

During the development of DECIMER Image Transformer (see Publication E), it became apparent that the application can partially recognise hand-written chemical structure depictions. Clévert et al. have reported similar observations for Img2Mol [84]. The goal to systematically assess the performance of DECIMER Image Transformer and other available tools on this type of structure depiction revealed the lack of a balanced benchmark dataset. ChemPix was published with a dataset of 600 hand-written structure depictions [120]. Still, this dataset has the substantial limitation of exclusively containing molecules consisting of carbon and hydrogen atoms. As a balanced dataset comprising 5088 manually drawn depictions of diversely picked molecules, the DECIMER Hand-Drawn Molecule Images dataset (see Publication D) fills a previously existing gap and represents a valuable contribution to the research in the field of OCSR.

3.5 Importance of diverse data for data-driven OCSR applications

The three main components of the DECIMER.ai system - DECIMER Segmentation, DECIMER Image Transformer and DECIMER Image Classifier - are entirely data-driven applications (see Publications A and E). That means they do not rely on hard-coded rules but learn to process information based on the training data during a supervised learning process. Large amounts of training data are necessary to exploit the full potential of modern deep-learning methods. While DECIMER Segmentation was trained using manually annotated data, the DECIMER Image Transformer and DECIMER Image Classifier models were trained on artificially generated datasets. For DECIMER Segmentation, a comparably small training data set of 9992 annotated chemical structures on 1820 pages was sufficient to yield the reported results (see Publication A). For the training of DECIMER Image Transformer and DECIMER Image Classifier, the training datasets were substantially more extensive, which would have made the usage of manually annotated data impossible. For example, the DECIMER Image Transformer model was trained on more than 450 million pairs of chemical structure depictions and the corresponding SMILES strings.

A common problem in the field of machine learning is the lack of applicability of trained machine learning systems on out-of-distribution (OOD) data [144]. This means that, for example, a deep-learning model may perform well on the exact type of data it has been trained on. Still, it may fail catastrophically once it is applied to data not described by the feature distribution of the training data. This logic can be applied to deep learning-based OCSR models as well. For example, the first published version of DECIMER Image Transformer performed well on depictions of molecules generated using the CDK with default depiction parameters [5]. Still, it failed on most real-world images from the literature. Similarly, the deep learning-based OCSR application SwinOCSR performs well on data similar to its training data [123] but fails in most real-world application scenarios (see Publication E).

Assuming there is a finite set of features that define how a chemical structure can be depicted, ideally, these features should all be represented in the training data of a deep learning-based OCSR tool to reduce the risk of failing when processing OOD samples and to achieve general applicability on real-world data. This problem is addressed by the chemical structure depiction generation tool RanDepict (see Publication B). By picking diverse sets of all available depiction parameters offered by four different cheminformatics toolkits, it aims to generate sets of depictions that represent the diversity of chemical structure depictions in the literature. A model trained on a sufficiently large diverse dataset should be capable of producing accurate results for a large variety of chemical structure depictions. The effectiveness of this training data diversification strategy can be demonstrated with the competitive results achieved with the recent version of DECIMER Image Transformer (see Publication E) that has been trained on chemical structure depictions generated using RanDepict. Especially the results on hand-written chemical structures demonstrate that the model trained on diverse data generalised relatively well on OOD samples as the training data did not contain a single hand-drawn chemical structure depiction. In the future, any deep learning-based model that is trained using pairs of images and string-based molecular structure representations can profit from the depiction generation with RanDepict.

Interestingly, the machine learning-based models and rule-based OCSR systems can be overfitted to a specific input data type. Clévert et al. have discovered that the rule-based OCSR-systems OSRA [95], Imago [96], and Molvec [97] perform well on the established benchmark datasets, but fail when slight image perturbations like minor rotations and shearing are introduced [84]. These results have been reproduced during the work on this dissertation (see Publication E). These OCSR systems mostly follow a predefined set of hard-coded rules. Still, they have a set of adjustable parameters that are likely overfitted to perform well on the available benchmark datasets. OSRA, for example, generates multiple molecular structures for each given image and uses an empirical confidence function to decide which of them to present as the final result [95].

Although the source code of Image2SMILES is not publicly available, Khokhlov et al. have published the training data generator. It pseudo-randomly picks different depiction parameters for generating chemical structure depictions, adds non-structural elements and other types of noise, and inserts different kinds of text labels that represent R-groups and functional groups [110]. The Image2SMILES data generator only uses one cheminformatics toolkit (RDKit). In contrast, RanDepict uses four (Chemistry Development Kit (CDK), RDKit, Indigo, Python-based Informatics Kit for Analysing Chemical Units (PIKAChU)) (see Publications B and E).

3.6 Importance of benchmark standards for OCSR

Besides generating OCSR training data, RanDepict can generate diverse test datasets representing various depiction features. This addresses the problem that the available benchmark datasets do not necessarily cover much of the chemical and depiction feature spaces. In many cases, the depicted molecules in the benchmark datasets are chemically similar and depicted similarly (Figure 12). Using diverse benchmark datasets with a large variety in the depiction feature distribution allows a lot more meaningful conclusions about the ability of an OCSR model to perform well on real-world data.

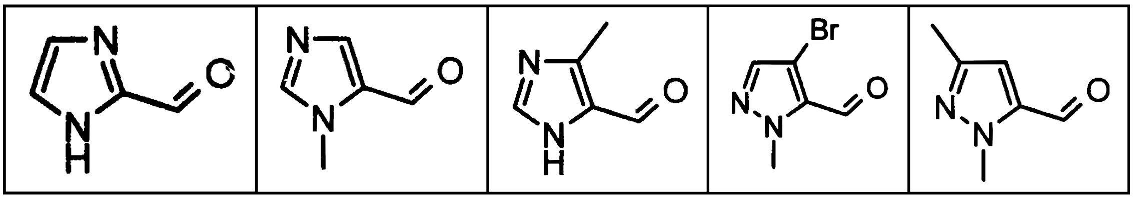


Figure 12: Samples of chemical structure depictions from a common OCSR benchmark dataset [101]

There is a lack of enforced benchmark standards in the field of OCSR. For example, the OCSR tool MolMiner has been benchmarked against the rule-based applications OSRA, Molvec and Imago on four independent test sets [125] while ignoring other deep learning-based methods like Img2Mol, DECIMER Image Transformer and SwinOCSR. The performance of the application Image2SMILES has only been evaluated in comparison to OSRA [110]. Besides selecting competitors, selecting favourable benchmark sets is another common phenomenon. For example, ABC-Net only uses one external benchmark dataset that contains mostly very clean chemical structure depictions without additional noise while ignoring all other available datasets [119]. Defining and enforcing clear standards in the field would be desirable to increase transparency and comparability of published performance results.

3.7 Contributions to other open projects

Apart from the projects designed explicitly for this dissertation, contributions to other open-source projects in the field were made to compare the performance of DECIMER Image Transformer to other OCSR tools (see Publication E). These are listed as examples of synergies in the open-source community. The compilation of OSRA [95] is a time-consuming process due to a large number of third-party dependencies [3]. To facilitate the tool's usage for anyone, a containerised [145] version of the tool was updated and published [146]. The initially published version of Img2Mol [84] sent a request containing the feature vector produced by the CNN encoder to a remote server which ran the CDDD decoder and returned the SMILES representation. To enable the processing of large batches of images without depending on a remote server, a locally running version of the CDDD decoder was integrated into the Img2Mol environment [117]. Additionally, an inference script has been added to the repository of SwinOCSR [124] as the tool was published without a script that enables running the model on an image to return a DeepSMILES string. These contributions are examples of synergies that can only occur due to the openly available source code of the applications. Similarly, all projects developed during the work described herein have the potential to profit from users' contributions as they are open-source projects.

3.8 The necessity of openly available chemical data

To fully profit from recent advances in the field of artificial intelligence - or, more specifically, machine learning - in chemistry, implementing and enforcing public FAIR data standards are crucial (see Publication D). In the era of data-driven applications, publishing chemical information exclusively in unstructured, human-readable data formats is an apparent inefficiency in the chemical research community. During the chemical research process, all information that is later published in an unstructured manner exists in a structured data format originally. For example, the results of nuclear magnetic resonance (NMR) spectroscopy experiments are collected in machine-readable, structured formats. Unfortunately, they are usually exclusively published in human-readable text or image formats. The need for chemical literature mining projects, as described in this dissertation, is caused by the inadequate publication habits of the research community. This is especially unsatisfying as there have been public calls for open data, open source, and open science (ODOSOS) in chemistry for nearly two decades [147, 148]. Machine learning has yielded significant contributions in areas of molecular informatics, where data has been available in structured formats (see Publication D). Additionally, there are clear perspectives that more areas will profit from FAIR and open data infrastructures in chemistry once they are in place [149].

Fortunately, there are initiatives that support FAIR data standards and open data repositories in chemistry. In Germany, a national research data infrastructure (Nationale Forschungsdateninfrastruktur, NFDI) is in the process of being implemented as a long-term project [150]. The chemistry consortium in the NFDI, NFDI4Chem, develops an electronic research data infrastructure for chemical data and supports the FAIR data principles [151]. For example, to avoid NMR data exclusively being published in text formats, the NFDI4Chem is developing nmrXiv, an open NMR data repository and analysis platform [152]. Another chemical research data platform is the Open Reaction Database (ORD), an open database for chemical reaction data [153]. With electronic research data infrastructure in place, the development of chemical literature mining applications described herein might become obsolete in the future. With the open availability of chemical data, the progress that has been achieved with data-driven, artificially intelligent applications in recent years (see Publication D) can be continued in other areas of chemistry.

All software packages and datasets developed as a part of this thesis are open-source projects published under permissive licences according to the FAIR standards. Other researchers or organisations can use and adapt them freely according to their specific needs. Due to the source code's open availability, users can raise issues publicly and contribute to the applications. This way, they have the potential to develop further based on community-driven suggestions and contributions.

4 Conclusion and outlook

Chemical structures can be represented in a variety of ways. Unfortunately, the representations published in the literature are commonly designed to be interpreted by humans and cannot easily be interpreted by a machine.

This work has contributed significantly to the automated extraction of chemical information from the scientific literature. Specifically, the field of OCSR has profited from open-source applications such as DECIMER Segmentation (see Publication A), DECIMER Image Transformer, DECIMER.ai (see Publication E) and RanDepict (see Publication B) as well as the Publication of the DECIMER Handwritten Image Dataset (see Publication C). Additionally, recent progress in molecular informatics enabled by the synergy of openly available data and AI-driven applications has been reviewed (see Publication D).

Segmenting chemical structures from whole pages is a fundamental step to automatically extracting chemical information from the literature. With DECIMER Segmentation, the first (and until today the only) deep learning-based open-source tool for the segmentation of chemical structures, has been published.

After the segmentation of chemical structure depictions, they need to be translated into a machine-readable representation of the underlying chemical graph. The version of the OCSR application DECIMER Image Transformer described in this work yields very competitive results in the presented comparative performance evaluation.

The success of the DECIMER Image Transformer would not have been possible without the training data diversification strategy implemented in the form of RanDepict. Like other encoder-decoder-based OCSR applications, DECIMER Image Transformer relies on an entirely data-driven approach where the model learns to interpret images with chemical structure depictions from the training data without any hard-coded rules. As the model exclusively learns to process information based on the training data, generating datasets containing diverse depiction features is the foundation of its excellent performance.

The integration of DECIMER Segmentation, DECIMER Image Classifier and DECIMER Image Transformer in the comprehensive automated chemical literature mining system DECIMER.ai enables users to automatically extract chemical structures from a chemical document in a graphical user interface. By embedding a molecular editor window, a human curator can assess and (if necessary) correct the results before saving them in a machine-readable format. DECIMER.ai is the first open platform that enables the automated extraction of chemical information from the literature.

In addition, the field of OCSR has profited from the publication of the DECIMER hand-drawn molecule image dataset. The dataset comprises diverse hand-drawn chemical structure depictions and is a valuable resource for evaluating the performance of OCSR applications for this specific type of depiction.

The applications described herein focus on extracting information from images in the scientific literature. In the future, these applications must be further extended to include text contents to enable a more complete information extraction. The integration of text-mining applications would enable to link information from the text to the extracted molecular structures. For example, analytical data is often exclusively described in the text. As the source code of all applications described herein is openly available, their functionalities can be extended by the research community in the future.

The automated literature mining systems developed during the work on this thesis can be integrated into AI-assisted submission pipelines of open databases. Chemical information can be automatically extracted from the literature to be made available in open databases. This way, the work presented herein represents a contribution to the open availability of chemical information in structured data formats.

References

[1] M. Krallinger, O. Rabal, A. Lourenço, J. Oyarzabal, A. Valencia, *Chem. Rev.* **2017**, *117*, 7673–7761.

[2] National Center for Biotechnology Information, PubChem Compound Summary for CID 2519, Caffeine. Retrieved April 20, 2023 from https://pubchem.ncbi.nlm.nih.gov /compound/Caffeine.

[3] K. Rajan, H. O. Brinkhaus, A. Zielesny, C. Steinbeck, *Journal of Cheminformatics* **2020**, *12*, 60.

[4] K. Rajan, A. Zielesny, C. Steinbeck, *Journal of Cheminformatics* **2020**, *12*, 65.

[5] K. Rajan, A. Zielesny, C. Steinbeck, *Journal of Cheminformatics* **2021**, *13*, 61.

[6] A. Dunning, M. De Smaele, J. Böhmer, *International Journal of digital curation* **1970**, *12*, 177–195.

[7] D. Rouvray, *Royal Institute of Chemistry Reviews* **1971**, *4*, 173–195.

[8] D. Janezic, A. Milicevic, S. Nikolic, N. Trinajstic, *Graph-theoretical matrices in chemistry*, CRC Press, **2015**.

[9] A. Dalby, J. G. Nourse, W. D. Hounshell, A. K. I. Gushurst, D. L. Grier, B. A. Leland, J. Laufer, *J. Chem. Inf. Comput. Sci.* **1992**, *32*, 244–255.

[10] R Panico, W. Powell, J.-C. Richer, *A guide to IUPAC Nomenclature of Organic Compounds*, *Vol. 2*, Blackwell Scientific Publications, Oxford, **1993**.

[11] T. Damhus, R. Hartshorn, A. Hutton, *Chemistry International* **2005**.

[12] L. Xu, P. Wu, S. J. Wright, L. Du, X. Wei, *J. Nat. Prod.* **2015**, *78*, 1841–1847.

[13] D. Weininger, *Journal of chemical information and computer sciences* **1988**, *28*, 31–36.

[14] S. R. Heller, A. McNaught, I. Pletnev, S. Stein, D. Tchekhovskoi, *Journal of Cheminformatics* **2015**, *7*, 23.

[15] I. Pletnev, A. Erin, A. McNaught, K. Blinov, D. Tchekhovskoi, S. Heller, *Journal of Cheminformatics* **2012**, *4*, 39.

[16] S. Heller, A. McNaught, S. Stein, D. Tchekhovskoi, I. Pletnev, *Journal of Cheminformatics* **2013**, *5*, 7.

[17] D. D. Palmer, *Handbook of natural language processing* **2000**, 11.

[18] M. Krenn, Q. Ai, S. Barthel, N. Carson, A. Frei, N. C. Frey, P. Friederich, T. Gaudin, A. A. Gayle, K. M. Jablonka, R. F. Lameiro, D. Lemm, A. Lo, S. M. Moosavi, J. M. Nápoles-Duarte, A. Nigam, R. Pollice, K. Rajan, U. Schatzschneider, P. Schwaller, M. Skreta, B. Smit, F. Strieth-Kalthoff, C. Sun, G. Tom, G. Falk von Rudorff, A. Wang, A. D. White, A. Young, R. Yu, A. Aspuru-Guzik, *Patterns* **2022**, *3*, 100588.

[19] N. O'Boyle, A. Dalke, *ChemRxiv* **2018**, DOI: 10.26434/chemrxiv.7097960.v1.

[20] M. Krenn, F. Häse, A Nigam, P. Friederich, A. Aspuru-Guzik, *arXiv preprint arXiv:1905.13741* **2019**.

[21] K. Rajan, C. Steinbeck, A. Zielesny, *Digital Discovery* **2022**, *1*, 84–90.

[22] W. A. Warr, *WIREs Comput Mol Sci* **2011**, *1*, 557–579.

[23] D. Teli, P. Balar, K. Patel, A. Sharma, V. Chavda, L. Vora, *Metabolites* **2023**, *13*, DOI `10.3390/metabo13020309`.

[24] D. S. Wigh, J. M. Goodman, A. A. Lapkin, *WIREs Computational Molecular Science* **2022**, *12*, e1603.

[25] A. Cereto-Massagué, M. J. Ojeda, C. Valls, M. Mulero, S. Garcia-Vallvé, G. Pujadas, *Methods* **2015**, *71*, Virtual Screening, 58–63.

[26] E. E. Bolton, Y. Wang, P. A. Thiessen, S. H. Bryant in (Eds.: R. A. Wheeler, D. C. Spellmeyer), Annual Reports in Computational Chemistry, Elsevier, **2008**, pp. 217–241.

[27] J. L. Durant, B. A. Leland, D. R. Henry, J. G. Nourse, *Journal of chemical information and computer sciences* **2002**, *42*, 1273–1280.

[28] D. Rogers, M. Hahn, *J. Chem. Inf. Model.* **2010**, *50*, 742–754.

[29] H. L. Morgan, *J. Chem. Doc.* **1965**, *5*, 107–113.

[30] M. J. McGregor, S. M. Muskal, *Journal of chemical information and computer sciences* **1999**, *39*, 569–74.

[31] F. Sandfort, F. Strieth-Kalthoff, M. Kühnemund, C. Beecks, F. Glorius, *Chem* **2020**, *6*, 1379–1390.

[32] L. Pattanaik, C. W. Coley, *Chem* **2020**, *6*, 1204–1207.

[33] D. K. Duvenaud, D. Maclaurin, J. Iparraguirre, R. Bombarell, T. Hirzel, A. Aspuru-Guzik, R. P. Adams, *Advances in neural information processing systems* **2015**, *28*.

[34] T. T. Tanimoto, *An elementary mathematical theory of classification and prediction by T.T. Tanimoto*, International Business Machines Corporation New York, **1958**, 10 p.

[35] L. R. Dice, *Ecology* **1945**, *26*, 297–302.

[36] A. Singhal, *IEEE Data Eng. Bull.* **2001**, *24*, 35–43.

[37] M. Ashton, J. Barnard, F. Casset, M. Charlton, G. Downs, D. Gorse, J. Holliday, R. Lahana, P. Willett, *Quantitative Structure-Activity Relationships* **2002**, *21*, 598–604.

[38] H. Wang, C. Ma, L. Zhou in 2009 international conference on information engineering and computer science, IEEE, **2009**, pp. 1–4.

[39] R. K. Dhanaraj, K. Rajkumar, U. Hariharan in *Business Intelligence for Enterprise Internet of Things*, (Eds.: A. Haldorai, A. Ramu, S. A. R. Khan), Springer International Publishing, Cham, **2020**, pp. 55–79.

[40] P. Cunningham, M. Cord, S. J. Delany in *Machine Learning Techniques for Multimedia: Case Studies on Organization and Retrieval*, (Eds.: M. Cord, P. Cunningham), Springer Berlin Heidelberg, Berlin, Heidelberg, **2008**, pp. 21–49.

[41] R. Nock, A. Menon in Proceedings of the 37th International Conference on Machine Learning, (Eds.: H. D. III, A. Singh), PMLR, **2020**, pp. 7370–7380.

[42] D. C. Montgomery, E. A. Peck, G. G. Vining, *Introduction to linear regression analysis*, John Wiley & Sons, **2021**.

[43] T. Hastie, J. Friedman, R. Tibshirani in *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*, Springer New York, New York, NY, **2001**, pp. 437–508.

[44] J. A. Hartigan, M. A. Wong, *Journal of the royal statistical society. series c (applied statistics)* **1979**, *28*, 100–108.

[45] L. P. Kaelbling, M. L. Littman, A. W. Moore, *Journal of artificial intelligence research* **1996**, *4*, 237–285.

[46] D. Silver, J. Schrittwieser, K. Simonyan, I. Antonoglou, A. Huang, A. Guez, T. Hubert, L. Baker, M. Lai, A. Bolton, et al., *nature* **2017**, *550*, 354–359.

[47] R. I. Mukhamediev, Y. Popova, Y. Kuchin, E. Zaitseva, A. Kalimoldayev, A. Symagulov, V. Levashenko, F. Abdoldina, V. Gopejenko, K. Yakunin, E. Muhamedijeva, M. Yelis, *Mathematics* **2022**, *10*, DOI `10.3390/math10152552`.

[48] F. Rosenblatt, *Psychological review* **1958**, *65*, 386.

[49] S. Chakraverty, D. M. Sahoo, N. R. Mahato in *Concepts of Soft Computing: Fuzzy and ANN with Programming*, Springer Singapore, Singapore, **2019**, pp. 183–188.

[50] L. N. Kanal in *Encyclopedia of Computer Science*, John Wiley and Sons Ltd., GBR, **2003**, 1383–1385.

[51] S. Sharma, S. Sharma, A. Athaiya, *Towards Data Sci* **2017**, *6*, 310–316.

[52] H. Ramchoun, M. J. Idrissi, Y. Ghanou, M. Ettaouil in Proceedings of the 2nd international Conference on Big Data, Cloud and Applications, **2017**, pp. 1–6.

[53] J. L. Elman, *Cognitive Science* **1990**, *14*, 179–211.

[54] Y. LeCun, Y. Bengio, G. Hinton, *Nature* **2015**, *521*, 436–444.

[55] M. Schuster, K. Paliwal, *IEEE Transactions on Signal Processing* **1997**, *45*, 2673–2681.

[56] Y. Bengio, P. Simard, P. Frasconi, *IEEE Transactions on Neural Networks* **1994**, *5*, 157–166.

[57] S. Hochreiter, J. Schmidhuber, *Neural computation* **1997**, *9*, 1735–1780.

[58] Y. Yu, X. Si, C. Hu, J. Zhang, *Neural computation* **2019**, *31*, 1235–1270.

[59] F. A. Gers, J. Schmidhuber, F. Cummins, *Neural computation* **2000**, *12*, 2451–2471.

[60] J. Chung, C. Gulcehre, K. Cho, Y. Bengio, Empirical Evaluation of Gated Recurrent Neural Networks on Sequence Modeling, **2014**.

[61] D. Bahdanau, K. Cho, Y. Bengio, *arXiv preprint arXiv:1409.0473* **2014**.

[62] P. Corbett, J. Boyle, *Journal of Cheminformatics* **2018**, *10*, 59.

[63] M. H. S. Segler, T. Kogej, C. Tyrchan, M. P. Waller, *ACS Cent. Sci.* **2018**, *4*, 120–131.

[64] K. Rajan, A. Zielesny, C. Steinbeck, *Journal of Cheminformatics* **2021**, *13*, 34.

[65] R. Winter, F. Montanari, F. Noé, D.-A. Clevert, *Chem. Sci.* **2019**, *10*, 1692–1701.

[66] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, I. Polosukhin, *Advances in neural information processing systems* **2017**, *30*.

[67] L. Tunstall, L. Von Werra, T. Wolf, *Natural language processing with transformers*, "O'Reilly Media, Inc.", **2022**.

[68] J. Devlin, M.-W. Chang, K. Lee, K. Toutanova, *arXiv preprint arXiv:1810.04805* **2018**.

[69] M. V. Koroteev, *CoRR* **2021**, *abs/2103.11943*.

[70] T. Brown, B. Mann, N. Ryder, M. Subbiah, J. D. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell, et al., *Advances in neural information processing systems* **2020**, *33*, 1877–1901.

[71] H. Touvron, T. Lavril, G. Izacard, X. Martinet, M.-A. Lachaux, T. Lacroix, B. Rozière, N. Goyal, E. Hambro, F. Azhar, et al., *arXiv preprint arXiv:2302.13971* **2023**.

[72] *Nature Biomedical Engineering* **2023**, *7*, 85–86.

[73] T. Isazawa, J. M. Cole, *J. Chem. Inf. Model.* **2022**, *62*, 1207–1213.

[74] S. Huang, J. M. Cole, *Chemical Science* **2022**, *13*, 11487–11495.

[75] T. Gupta, M. Zaki, N. M. A. Krishnan, Mausam, *npj Computational Materials* **2022**, *8*, 102.

[76] P. Schwaller, T. Laino, T. Gaudin, P. Bolgar, C. A. Hunter, C. Bekas, A. A. Lee, *ACS central science* **2019**, *5*, 1572–1583.

[77] P. Schwaller, A. C. Vaucher, T. Laino, J.-L. Reymond, *Machine learning: science and technology* **2021**, *2*, 015016.

[78] P. Schwaller, R. Petraglia, V. Zullo, V. H. Nair, R. A. Haeuselmann, R. Pisoni, C. Bekas, A. Iuliano, T. Laino, *Chemical science* **2020**, *11*, 3316–3325.

[79] D. Dai in 2021 International Conference on Big Data, Artificial Intelligence and Risk Management (ICBAR), **2021**, pp. 135–138.

[80] K. Fukushima, *Neural networks* **1988**, *1*, 119–130.

[81] S. Albawi, T. A. Mohammed, S. Al-Zawi in 2017 International Conference on Engineering and Technology (ICET), **2017**, pp. 1–6.

[82] N. Akhtar, U. Ragavendran, *Neural Computing and Applications* **2020**, *32*, 879–898.

[83] K. Xu, J. Ba, R. Kiros, K. Cho, A. Courville, R. Salakhudinov, R. Zemel, Y. Bengio in Proceedings of the 32nd International Conference on Machine Learning, (Eds.: F. Bach, D. Blei), PMLR, Lille, France, **2015**, pp. 2048–2057.

[84] D.-A. Clevert, T. Le, R. Winter, F. Montanari, *Chem. Sci.* **2021**, *12*, 14174–14181.

[85] R. Aggarwal, A. Gupta, V. Chelur, C. V. Jawahar, U. D. Priyakumar, *J. Chem. Inf. Model.* **2022**, *62*, 5069–5079.

[86] I. Lauriola, A. Lavelli, F. Aiolli, *Neurocomputing* **2022**, *470*, 443–456.

[87] L. A. Miccio, G. A. Schwartz, *Polymer* **2020**, *193*, 122341.

[88] M. Hirohara, Y. Saito, Y. Koda, K. Sato, Y. Sakakibara, *BMC Bioinformatics* **2018**, *19*, 526.

[89] M. Tan, Q. Le in International conference on machine learning, PMLR, **2021**, pp. 10096–10106.

[90] A. M. Hafiz, G. M. Bhat, *International Journal of Multimedia Information Retrieval* **2020**, *9*, 171–189.

[91] K. He, G. Gkioxari, P. Dollár, R. Girshick in 2017 IEEE International Conference on Computer Vision (ICCV), **2017**, pp. 2980–2988.

[92] F. Musazade, N. Jamalova, J. Hasanov, *Journal of Cheminformatics* **2022**, *14*, 61.

[93] J. R. McDaniel, J. R. Balmuth, *J. Chem. Inf. Comput. Sci.* **1992**, *32*, 373–378.

[94] J. Staker, K. Marshall, R. Abel, C. M. McQuaw, *J. Chem. Inf. Model.* **2019**, *59*, 1017–1029.

[95] I. V. Filippov, M. C. Nicklaus, *J. Chem. Inf. Model.* **2009**, *49*, 740–743.

[96] V. Smolov, F. Zentsev, M. Rybalkin in TREC, Citeseer, **2011**.

[97] MOLVEC: Open source library for chemical structure recognition, ACS Meeting, **2019**.

[98] P Ibison, M Jacquot, F Kam, A. Neville, R. W. Simpson, C Tonnelier, T Venczel, A. P. Johnson, *Journal of Chemical Information and Computer Sciences* **1993**, *33*, 338–344.

[99] A. T. Valko, A. P. Johnson, *J. Chem. Inf. Model.* **2009**, *49*, 780–787.

[100] M. Zimmermann in Twentieth Text REtrieval Conference (TREC 2011) Proceedings. **2011**.

[101] N. M. Sadawi, A. P. Sexton, V. Sorge in 19th Document Recognition and Retrieval Conference (DRR 2012), (Eds.: C. Viard-Gaudin, R. Zanibbi), SPIE, **2012**.

[102] J. Park, G. R. Rosania, K. A. Shedden, M. Nguyen, N. Lyu, K. Saitou, *Chemistry Central Journal* **2009**, *3*, 4.

[103] P. Frasconi, F. Gabbrielli, M. Lippi, S. Marinai, *J. Chem. Inf. Model.* **2014**, *54*, 2380–2390.

[104] D. Pavlov, M. Rybalkin, B. Karulin, M. Kozhevnikov, A. Savelyev, A Churinov, *Journal of cheminformatics* **2011**, *3*, P4.

[105] O. Ronneberger, P. Fischer, T. Brox in Medical Image Computing and Computer-Assisted Intervention–MICCAI 2015: 18th International Conference, Munich, Germany, October 5-9, 2015, Proceedings, Part III 18, Springer, **2015**, pp. 234–241.

[106] J. Long, E. Shelhamer, T. Darrell in Proceedings of the IEEE conference on computer vision and pattern recognition, **2015**, pp. 3431–3440.

[107] M. Oldenhof, A. Arany, Y. Moreau, J. Simm, *J. Chem. Inf. Model.* **2020**, *60*, 4506–4517.

[108] ChemGrapher (GitHub), https://github.com/biolearning-stadius/chemgrapher-self-rich-labeling, Accessed: 20th March 2023, **2021**.

[109] K. He, X. Zhang, S. Ren, J. Sun in Proceedings of the IEEE conference on computer vision and pattern recognition, **2016**, pp. 770–778.

[110] I. Khokhlov, L. Krasnov, M. V. Fedorov, S. Sosnin, *Chemistry–Methods* **2022**, *2*, e202100069.

[111] Img2SMILES$_{g}enerator(GitHub)$, https://github.com/syntelly/img2smiles$_{g}enerator$, Accessed: 20th March 2023, **2021**.

[112] M. Tan, Q. Le in Proceedings of the 36th International Conference on Machine Learning, (Eds.: K. Chaudhuri, R. Salakhutdinov), PMLR, **2019**, pp. 6105–6114.

[113] E. L. Willighagen, J. W. Mayfield, J. Alvarsson, A. Berg, L. Carlsson, N. Jeliazkova, S. Kuhn, T. Pluskal, M. Rojas-Chertó, O. Spjuth, G. Torrance, C. T. Evelo, R. Guha, C. Steinbeck, *Journal of Cheminformatics* **2017**, *9*, 33.

[114] DECIMER Image Transformer, https://github.com/Kohulan/DECIMER-Image$_Transformer$, Accessed: 16th March 2023.

[115] RDKit: Open-source cheminformatics, `http://www.rdkit.org`, Accessed: 11th April 2023.

[116] OEChem-TK, OpenEye Scientific Software, http://www.eyesopen.com.

[117] Img2Mol (Github), https://github.com/bayer-science-for-a-better-life/Img2Mol, Accessed: 15th March 202, **2021**.

[118] MICER (GitHub), https://github.com/Jiacai-Yi/MICER, Accessed: 16th March 2023.

[119] X.-C. Zhang, J.-C. Yi, G.-P. Yang, C.-K. Wu, T.-J. Hou, D.-S. Cao, *Briefings in Bioinformatics* **2022**, *23*, DOI `10.1093/bib/bbac033`.

[120] H. Weir, K. Thompson, A. Woodward, B. Choi, A. Braun, T. J. Martínez, *Chem. Sci.* **2021**, *12*, 10622–10633.

[121] ChemPix (GitHub), https://github.com/mtzgroup/ChemPixCH, Accessed: 16th March 2023.

[122] Z. Liu, Y. Lin, Y. Cao, H. Hu, Y. Wei, Z. Zhang, S. Lin, B. Guo, Swin Transformer: Hierarchical Vision Transformer using Shifted Windows, **2021**.

[123] Z. Xu, J. Li, Z. Yang, S. Li, H. Li, *Journal of Cheminformatics* **2022**, *14*, 41.

[124] SwinOCSR (GitHub), https://github.com/suanfaxiaohuo/SwinOCSR, Accessed: 15th March 2023, **2022**.

[125] Y. Xu, J. Xiao, C.-H. Chou, J. Zhang, J. Zhu, Q. Hu, H. Li, N. Han, B. Liu, S. Zhang, J. Han, Z. Zhang, S. Zhang, W. Zhang, L. Lai, J. Pei, *J. Chem. Inf. Model.* **2022**, *62*, 5321–5328.

[126] Y. Qian, J. Guo, Z. Tu, Z. Li, C. W. Coley, R. Barzilay, *J. Chem. Inf. Model.* **2023**, *63*, 1925–1934.

[127] MolScribe (GitHub), https://github.com/thomas0809/MolScribe, Accessed: 16th March 2023.

[128] B. R. Terlouw, S. P. J. M. Vromans, M. H. Medema, *Journal of Cheminformatics* **2022**, *14*, 34.

[129] L. Chen, G. Papandreou, F. Schroff, H. Adam, *CoRR* **2017**, *abs/1706.05587*.

[130] E. Beard, J. M. Cole, *J. Chem. Inf. Model.* **2020**, DOI `10.1021/acs.jcim.0c00042`.

[131] D. M. Wilary, J. M. Cole, *J. Chem. Inf. Model.* **2021**, *61*, 4962–4974.

[132] J. Sander, M. Ester, H.-P. Kriegel, X. Xu, *Data Mining and Knowledge Discovery* **1998**, *2*, 169–194.

[133] B. Karulin, M. Kozhevnikov, *Journal of Cheminformatics* **2011**, *3*, P3.

[134] M. C. Swain, J. M. Cole, *J. Chem. Inf. Model.* **2016**, *56*, 1894–1904.

[135] J. Mavračić, C. J. Court, T. Isazawa, S. R. Elliott, J. M. Cole, *J. Chem. Inf. Model.* **2021**, *61*, 4280–4289.

[136] J. Devlin, M.-W. Chang, K. Lee, K. Toutanova, BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding, **2018**.

[137] C. J. Court, J. M. Cole, *Scientific data* **2018**, *5*, 180111.

[138] S. Huang, J. M. Cole, *Scientific Data* **2020**, *7*, 1–13.

[139] E. J. Beard, G. Sivaraman, Vázquez-Mayagoitia, V. Vishwanath, J. M. Cole, *Scientific Data* **2019**, *6*, 307.

[140] J. Zhao, J. M. Cole, *Scientific Data* **2022**, *9*, 192.

[141] Q. Dong, J. M. Cole, *Scientific Data* **2022**, *9*, 193.

[142] E. J. Beard, J. M. Cole, *Scientific Data* **2022**, *9*, 329.

[143] O. Sierepeklis, J. M. Cole, *Scientific Data* **2022**, *9*, 648.

[144] D. Berend, X. Xie, L. Ma, L. Zhou, Y. Liu, C. Xu, J. Zhao in Proceedings of the 35th IEEE/ACM International Conference on Automated Software Engineering, Association for Computing Machinery, Virtual Event, Australia, **2021**, 1041–1052.

[145] J. Turnbull, *The Docker Book: Containerization is the new virtualization*, James Turnbull, **2014**.

[146] H. O. Brinkhaus, docker-osra, https://hub.docker.com/r/obrink/osra, Accessed: 15th March 2023, **2022**.

[147] R. Guha, M. T. Howard, G. R. Hutchison, P. Murray-Rust, H. Rzepa, C. Steinbeck, J. Wegner, E. L. Willighagen, *Journal of Chemical Information and Modeling* **2006**, *46*, Cited by: 366; All Open Access, Green Open Access, Hybrid Gold Open Access, 991 – 998.

[148] N. M. O'Boyle, R. Guha, E. L. Willighagen, S. E. Adams, J. Alvarsson, J.-C. Bradley, I. V. Filippov, R. M. Hanson, M. D. Hanwell, G. R. Hutchison, C. A. James, N. Jeliazkova, A. S. I. D. Lang, K. M. Langner, D. C. Lonie, D. M. Lowe, J. Pansanel, D. Pavlov, O. Spjuth, C. Steinbeck, A. L. Tenderholt, K. J. Theisen, P. Murray-Rust, *Journal of Cheminformatics* **2011**, *3*, 37.

[149] M. Scheffler, M. Aeschlimann, M. Albrecht, T. Bereau, H.-J. Bungartz, C. Felser, M. Greiner, A. Groß, C. T. Koch, K. Kremer, W. E. Nagel, M. Scheidgen, C. Wöll, C. Draxl, *Nature* **2022**, *604*, 635–642.

[150] N. Hartl, E. Wössner, Y. Sure-Vetter, *Informatik Spektrum* **2021**, *44*, 370–373.

[151] C. Steinbeck, O. Koepler, F. Bach, S. Herres-Pawlis, N. Jung, J. Liermann, S. Neumann, M. Razum, C. Baldauf, F. Biedermann, et al., *Research ideas and outcomes* **2020**, *6*, e55852.

[152] NFDI4Chem, nmrXiv - Open, FAIR and Consensus-Driven NMR spectroscopy data repository and analysis platform, https://nmrxiv.org/, Accessed: 07th April 2023, **2023**.

[153] S. M. Kearnes, M. R. Maser, M. Wleklinski, A. Kast, A. G. Doyle, S. D. Dreher, J. M. Hawkins, K. F. Jensen, C. W. Coley, *J. Am. Chem. Soc.* **2021**, *143*, 18820–18826.

Declarations

Selbstständigkeitserklärung

Ich erkläre, dass ich die vorliegende Arbeit selbständig und unter Verwendung der angegebenen Hilfsmittel, persönlichen Mitteilungen und Quellen angefertigt habe.

....	....
(Ort, Datum)	(Unterschrift des Verfassers)

Erklärung zu den Eigenanteilen des Promovenden sowie der weiteren Doktoranden/Doktorandinnen als Koautoren an den Publikationen und Zweitpublikationsrechten bei einer kumulativen Dissertation

Alle veröffentlichten Manuskripte, die in dieser kumulativen Dissertation verwendet werden, sind in Open-Access-Zeitschriften publiziert, die eine uneingeschränkte Nutzung, Verbreitung und Vervielfältigung in jedem Medium erlauben, sofern das Originalwerk ordnungsgemäß zitiert wird ("Reprint Permissions"). Die Koautoren der in dieser kumulativen Dissertation verwendeten Manuskripte sind sowohl uber die Nutzung, als auch uber die obenangegebenen Eigenanteile der weiteren Doktoranden/Doktorandinnen als Koautorenan den Publikationen und Zweitpublikationsrechten bei einer kumulativen Dissertationinformiert und stimmen diesem zu. Die Anteile des Promovenden sowie der weiteren Doktoranden/Doktorandinnen und Koautoren an den Publikationen und Zweitpublikationsrechten bei einer kumulativen Dissertation sind der jeweiligen Publikation vorangestellt.

....

(Ort, Datum) (Unterschrift des Verfassers)

Einverständniserklärung der Betreuenden zur kumulativen Dissertation

Ich bin mit der Abfassung der Dissertation als publikationsbasierte Dissertation, d.h. kumulativ, einverstanden und bestätige die vorstehenden Angaben.

....

(Ort, Datum) (Prof. Dr. Christoph Steinbeck)

Acknowledgements

This work would not have been possible without the help of many people. I consider myself very lucky to have had the opportunity to work in this exciting field with so many great people. I am incredibly grateful for all the support that has led to me being able to hand up this thesis. I think it is futile to try to express the enormous amount of gratitude that I feel in an adequate way, but it would be wrong not to make an attempt.

I would first like to thank Prof. Dr. Christoph Steinbeck for his guidance and supervision during my work. I think I could not imagine better conditions than what I found in your group while diving into a new and exciting field. You exude calmness and kindness, and that has a positive influence on the atmosphere of your whole environment. Whenever I asked, I have received help and guidance, but I have not felt negative pressure a single time. I have had the freedom to run after ideas and dive into new concepts and technologies to eventually contribute to something of value for the community. I am incredibly grateful for having gotten this opportunity.

Speaking of people who exude calmness and kindness - I would also like to thank Prof. Dr. Achim Zielesny for his guidance and supervision. Achim, your excitement and your enthusiasm are contagious. You always come up with good ideas, and I truly appreciate your input, even if I might not want to hear it at first because it means more work for me. It is a true pleasure to work with you. Thank you for the assistance and all the good conversations we had over the past years.

Another person without whom this work could not have been what it is today is my friend and colleague, Dr. Kohulan Rajan. I am very happy that years ago, I started as your student assistant on the curation of data for a project that would later turn into DECIMER Segmentation. I love our discussions about technology and how passionate you are about things. It is an absolute joy to work with you, and I could not imagine a better person to work with.

I would also like to thank Jonas Schaub. You are not just always friendly and fun to be around, you also are the only reason why the plants in the upstairs office have not dried out in a desert created by your co-workers. But honestly, thank you for being you and for always having an open ear for every kind of problem. I admire your willingness to engage for justice and a better world. Your attention to detail and your thoroughness lead to a better result in more or less anything you are involved in.

I would like to acknowledge Mahnoor Zulfiqar. Although we don't work on many projects together, it's a pleasure to share an office with you. You are always kind, friendly, and approachable, which creates a positive and welcoming atmosphere in the upstairs office. I could not wish for better co-workers, and you are an essential reason things work out so well in our group.

Although she left the group a while ago, I would like to take a moment to acknowledge

Adelene Lai for the positive impact she has made on our workplace. I have really enjoyed our discussions in the office. Thanks for being an amazing colleague!

Another person who I consider to be an inspiration is Dr. Maria Sorokina. Maria, you have been missed a lot since you left us! Your positive energy has really brightened up the office, and I have always enjoyed our discussions.

I am in a very lucky position and have the honour of working with a group of absolutely lovely and supportive people. Thanks to everyone from the *Cheminformatics and Computational Metabolomics* group - Anne Abel, Christian Popp, Felix Bänsch, Franziska Eberl, Jonas Dietrich, Luiz Gadelha, Maria Isabel Agea, Michael Wenk, Nisha Sharma, Noura Rayya, Sarah Tippner, Vinay Singh, Viktor Weißenborn and Venkata Chandrasekhar Nainala (Chandu).

I would like to express my deepest gratitude to my family for their support and encouragement throughout my journey. I would like to thank my mother Rosa-Maria Tebroke, whose attention to detail has inspired me to pursue excellence. With his analytical mind, my father Joachim Brinkhaus has taught me to approach problems in a logical and systematic way. From my step-father Wilhelm Pazdera's calmness and level-headedness, I have learned to approach difficult situations with a clear and rational mind. My brother Jan Brinkhaus is an example of successful entrepreneurship. My sister Beke Brinkhaus is an impressive example of perseverance in her pursuit of academic education. I would also like to thank my brother Maximilian Brinkhaus who is a real inspiration for me and who never gets tired of giving good advice. I also want to thank John and Breda Brady, whose generosity and kindness have made a significant impact on my personal growth and development. Moreover, I would like to acknowledge Eoghan Brady for being an incredible brother-in-law who has enriched my life with his intelligence and kindness.

I would have to write another book full of acknowledgements if I wanted to properly thank my wife Aoife Brady. You are the love of my life. You support me in everything I do. I cannot put into words how excited I am about our future together.

Thank you for everything; I could not have done it without you.