IMPLEMENTING TRANSFER LEARNING FOR MITOTIC CELL DETECTION

An Undergraduate Research Scholars Thesis

by

MUHAMMED ZAHID KAMIL

Submitted to the Undergraduate Research Scholars program at
Texas A&M University
in partial fulfillment of the requirements for the designation as an

UNDERGRADUATE RESEARCH SCHOLAR

Approved by Research Advisor: Dr. Jim Ji

May 2020

Major: Electrical Engineering

TABLE OF CONTENTS

ABSTRACT

Implementing Transfer Learning for Mitotic Cell Detection

Muhammed Zahid Kamil
Department of Electrical and Computer Engineering
Texas A&M University

Research Advisor: Dr. Jim Ji
Department of Electrical and Computer Engineering
Texas A&M University

The primary objective of this project is to use a neural network (deep learning) model that will be trained on datasets of human cells in mitosis and then apply it on animal cells using transfer learning techniques. The purpose of this project is to speed up the process for the pathologist to detect mitotic activity for cancer diagnosis and determine transfer learning techniques that will aid in predicting mitotic activity in animal cells. This information will prove to be useful for correct diagnosis of cancer and reduce potential errors of detecting mitotic cells. Instead of acquiring the highly meticulous task of the labelled data, transfer learning can produce promising results by reusing the learned features from human datasets and applying the learned knowledge to the datasets of the animal cells. This project proposes to develop a higher detection accuracy with our framework of transfer learning and optimize the deep learning model to produce the desired output of detecting mitotic cells in animals with the limited amount of training data. Transfer learning techniques will be aimed to improve the performance of learning in other domains and reduce the cost of the expensive labeled data. Preliminary system will be tested and analyzed to gain insights. Due to limited time and resource, the project only tests on publicly available human datasets.

DEDICATION

I would like to dedicate this paper to my family and friends. A special gratitude towards my loving parents Faiz Kamil and Najma Kamil who have supported me all these years. I would also like to thank my sister Zahra Kamil and my two brothers Zaid and Zafar Kamil.

A special thank you to my university Texas A&M University at Qatar, Qatar Foundation, student housing and the state of Qatar for providing me with this opportunity to be have an education.

ACKNOWLEDGMENTS

I would like to thank my faculty advisor Dr. Jim Ji and research assistant Ms. Imene Mecheter for their guidance and support throughout the course of this research.

Special thanks to Ms. Imene Mecheter for her specific contributions in obtaining the datasets, preparing the data and sharing her algorithms in this research.

Thanks also goes to my friends and colleagues from the Electrical and Computer Engineering department faculty and staff for making my time at Texas A&M University a great experience. I also want to extend my gratitude to Qatar Foundation and the people of Qatar for accepting and aiding me in becoming an educated individual.

NOMENCLATURE

CNN – Convolutional Neural Network

ICPR12 – International Conference on Pattern Recognition 2012

ReLU – Rectified Linear Unit

ANN – Artificial Neural Network

FCN – Fully Connected Network

VGG-16 – Visual Geometry Group 2016

CHAPTER I

INTRODUCTION

According to the World Health Organization (WHO) breast cancer is one of the leading types of cancer in the world that has affected approximately 2.1 million women each year [9]. This year, an estimated number of 41,760 deaths from women will occur [10]. The main goal in cancer diagnosis is to identify breast cancer at early stages. This allows for effective treatment and reduces the number of risks of death from breast cancer. For the grading of breast cancer, pathologists use the Bloom and Richardson system to assign a score from 1 to 3 of the tissue sample by summing up the following characteristics: presence of glandular/tubular structures, nuclear pleomorphism and mitotic count ([4],[5]). The problem arises when pathologists undergo the cumbersome mitotic counting task to determine the stage of cancer. This project aims to speed up the detection task and the need for manually labelled data by applying a neural network trained on Human cells to animal cells for mitotic detection.

Mitotic counting refers to counting the number of cells undergoing mitosis (cell division). The different phases of mitosis (prophase, metaphase, anaphase and telophase) indicates that the cell is undergoing mitosis. The number and density of mitotic cells is to indicate the aggressiveness of the tumor to classify the stage and grade of cancer in the patient.

The access to public datasets for human histopathological images in the detection of breast cancer supports the development of incorporating the deep learning model for image classification ([4],[5]). For example, the public datasets from the MITOS-ATYPIA-2014[6], AMIDA13[7] and TUPAC16[8] challenges have been used to generate deep learning models to improve accuracy of mitotic detection in the image samples for human cells.

However, with the publicly available datasets of human histopathological images, very little has been done to automatically detect mitotic cells in animal histopathological images. The prospect of transferring the pre-trained neural network model from human to animal histopathological images can potentially eradicate the problem of the lack of labelled images, limited data and weak annotations [3] from pathologists that hinders the development of a powerful model. This project will use two datasets, ICPR12-2014 and animal histological images from Texas A&M Veterinary Medical Diagnostic Laboratory (TVMDL). The dataset for human cells were stained using hematoxylin and eosin (H&E) that stains the nuclei in a purple-blue shade and the cytoplasm ranging from a reddish to a pinkish color [4].

This project will use convolutional neural networks (CNNs) to train the deep learning model as it has proved to perform better than classic machine learning techniques on image classification [4]. However, the performance of neural networks is determined by the quality of the training datasets and have been shown to produce undesirable outputs such as identifying empty space in images as dead cells or recognizing patterns in the order of the training datasets but not in the images themselves [3]. This project will also aim to minimize the error output of the deep-learning model and determine a minimum amount of data for transfer learning in future applications.

The overall objective of this project is to develop an automatic tool for detecting mitotic cells from histopathological animal images for preclinical studies. The goal of transfer learning is to apply knowledge from previously learned tasks and apply them to newer tasks that do not have enough training datasets to develop a deep-learning model. With transfer learning, the trained model can be fine-tuned into automating similar tasks and thus eliminate the need for

large datasets and accelerate the learning process in comparison to traditional Machine learning techniques ([3],[4]).

CHAPTER II

METHODS

Vision is a huge part of human life. It enables us to see, learn and emulate from others. Thanks to Convolutional Neural Networks (CNNs), computer vision is allowed as computers can now 'see' but in a completely different way as to humans. Computers see images by reading them in the form of pixels which are just values ranging from 0 to 255. To the computer an image is just an array of numbers and each object has its own pattern and this is what the computer will use to recognize an image. The famous applications of CNNs are for medical care, autonomous cars, facial recognition, Natural Language Processing applications etc. [13].

Preprocessing

The figures below show how the input is generated for the neural network. The training data are figures 1 and 2 while the neural network will learn from the binary labelled image (figure 3) and start using predictive analysis to locate mitotic cell activity.

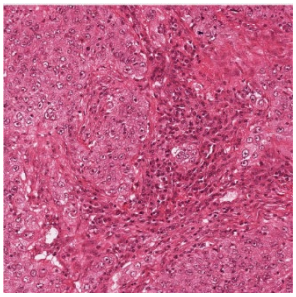


Figure 1: ICPR12 histopathological image

In figure 2, the labelled data can be seen as yellow markings on different areas on the image. These marking were done manually done by a pathologist in identifying the mitotic cell activity. The binary labelled data shown in figure 3 are the results of the morphological

operations [1] such as subtracting and closing to show the location of the mitotic activity. This binary labelled data are masks showing the locations of mitotic cells and this is known as the ground truth that will be fed along with the original image shown in figure 1. The dataset consists of 35 images of size of 2084x2084x3 pixels. We create patches of 96x96x3 to feed into the CNN architecture

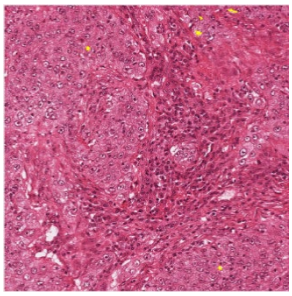


Figure 2: ICPR12 labelled image

Figure 3: Binary labelled Images

Convolutional Neural Network

A Convolutional Neural Network is a Deep Learning Algorithm that is made up of complex feedforward neural networks and is used mainly for image classification because of its high accuracy. It is inspired by the organization of the animal visual cortex, where there are small regions of cells sensitive to specific of visual field. Some of these cells are sensitive to

vertical/horizontal edges and is similar to the outputs of passing through the first few Convolutional layers [23].

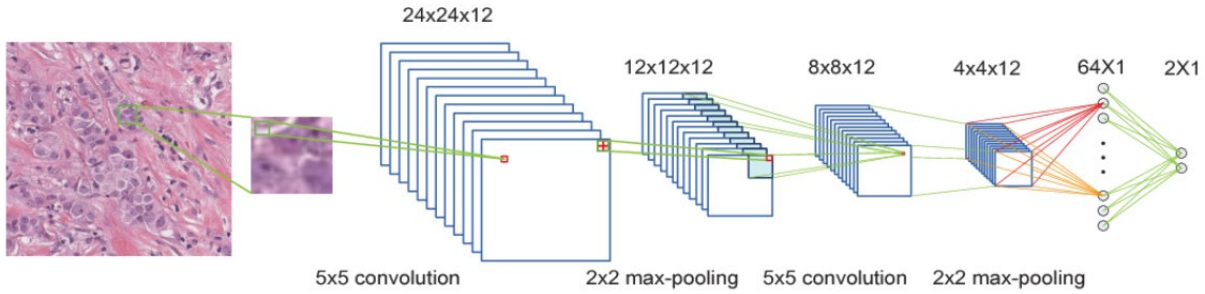


Figure 4: A typical Convolutional Neural Network architecture [11]

There are many aspects in a CNN and the most important characteristics, also shown in figure 4 are: Convolutional layers, kernel filter, strides, pooling, padding, activation functions, fully connected networks, softmax function, feature maps, dropout, optimizer, kernel initializer, depth, convolution and convolution transpose.

The input layer accepts a size of the image height x image width x depth (normally 3 for RGB color channels). The picture is made up of pixel values that range from 0 to 255. Therefore, the input matrix is a matrix of numbers ranging from (0,0,0) to (255,255,255) corresponding to black and white respectively.

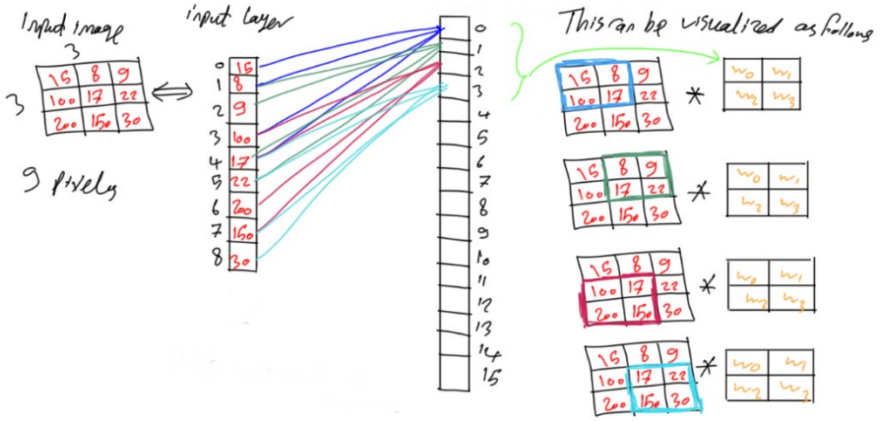


Figure 5: Convolution with kernel of size 2x2 with input size of 3x3 [21]

A convolutional layer produces a **feature map** through using the linear convolution operation between the input matrix or resulting feature matrix and the **kernel filter**. The size of the kernel filter (where its size is smaller than the size of the feature maps) and its process is shown in figure 5 and 6. The kernel filter slides across the input matrix. The problem lies when the size of the kernel matrix does not cover all the elements in the input matrix. Therefore, we add zeros symmetrically around the border region of the input matrix, which is called **padding**. Padding is also called pixel adding and is used to prevent the loss of important features and resolution especially for reconstructing the image such as in this case.

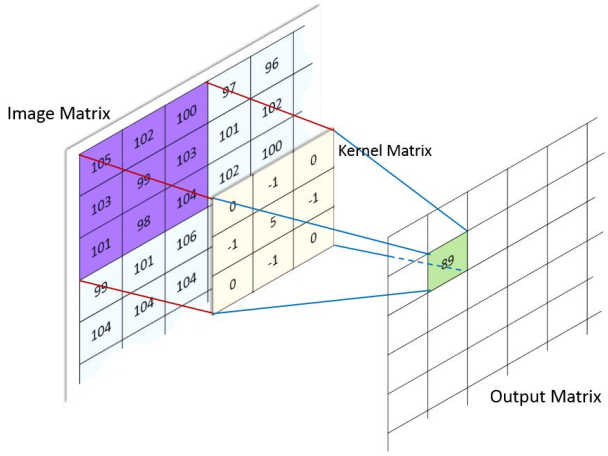


Figure 6: Convolution operation with kernel to produce feature map [20]

The **depth** in this case is considered to be the number of the independent kernel filters which also corresponds to producing to the same number of feature maps. The **stride** is the number of positions the kernel would move at the input matrix to do the convolution operation. For example, the stride is said to be 1 if the kernel moves over the matrix 1 pixel at a time [14]. This is so far the 1st convolutional layer. For the 2nd convolutional layer, the output of the first convolutional layer becomes the input of the 2nd convolutional layer. To calculate the dimensions

of the output using a specific size of a kernel, padding and stride we use the formula below in equation 1 and the explanation of the variables are given in figure 7 below.

$$W_2 = \frac{W_1 - F + 2P}{S} + 1 \quad (1)$$

W_1 – is the width/height of the input tensor
F – is the width/height of the kernel
P – is the padding
S – is the stride
W_2 – is the output /height

Figure 7: Formula to calculate the dimension of the feature map [20]

However, all of this is a linear operation and we need to introduce some non-linearity into the learned feature maps. Non-linear **activation functions** are important and most non-linear function are used between layers of neural networks. Non-linear activation functions are preferred to satisfy the following [22]:

- Can approximate any function (layered neural network)
- Continuously differentiable that allows for backpropagation
- Functions to have a finite range to lead to stable performance or else it may be a heavy computation load if it extends to infinitesimal values
- Smooth functions are preferred
- Behave like identity functions near the origin where f(x) = x

However, most Convolutions prefer the **Rectified Linear Unit function** (ReLU), shown below in figure 8, because it is shown to produce large and consistent gradients that is useful for

gradient descent and backpropagation [22]. It also allows it to converge to any function and reduces the calculation load compared to the other activation function such as sigmoid and hyperbolic tangent functions [18].

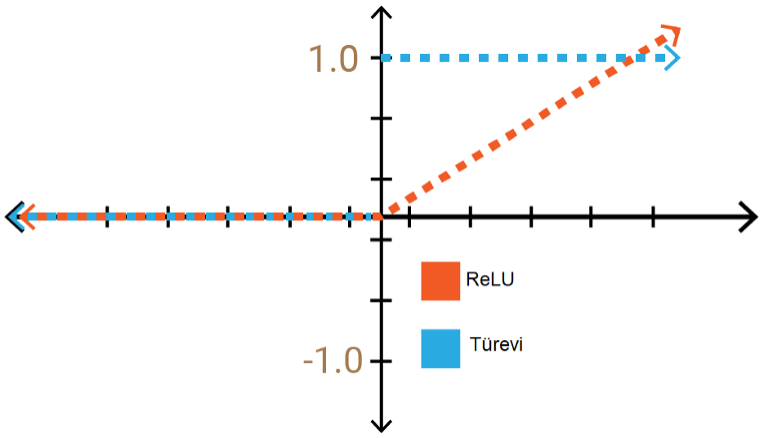


Figure 8: ReLU function in orange [18]

Pooling is used to reduce the dimensionality of the feature map i.e. down sample. Pooling layers, follow in sequence to convolutional layers. These layers are known to aid in the translation invariance for image processing [22]. Translation invariance is recognizing an object as an object even when the object is varied in some way as it preserves the objects identity. More specifically, it means that each point/pixel has moved the same amount in the same direction [24]. This is very essential in localization of the feature as we use this to recognize the mitotic activity in the cells. Pooling also helps reduce overfitting and reduces the computational load. Pooling downsamples the image in its height and width but keeps the depth/number of the channels the same [20]. This project uses **max pooling** as the pooling operation. The operation is shown in figure 9 below where it is shown to downsample by taking the maximum pixel value of an input size defined by the max pooling size.

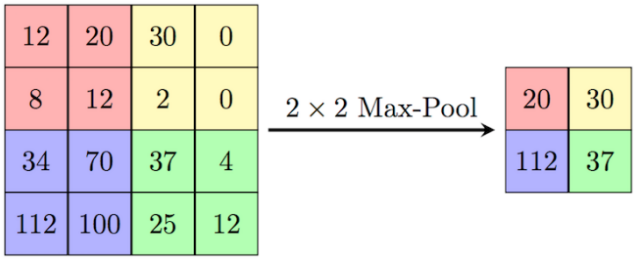


Figure 9: Max pooling [20]

The **Fully Connected Layers** (FCNs) is used to extract the high-level features. FCNs are just simple feedforward neural networks as shown in figure 10 below [16]. These feedforward neural networks or Artificial Neural Networks (ANNs) are fully connected where each neuron is connected to every other neuron in the next layer. Each neuron is multiplied by a weight w_{ij} and an added bias b_j and then passed to the next layer of neurons. The typical formula for each neuron is given in figure 11 below. The activation function f in the hidden layers is normally a ReLU function. ReLU functions help with the vanishing gradient problem when backpropagating [18]. The last function 'g' in figure 8 that is connected to the output layer is most probably a **softmax** (shown below in figure 12), function for a CNN. This is a common activation function is used for multi-class classification in CNNs as it outputs a vector representing the probability distributions [25].

The output is then based on the class with the maximum probability also known as the maximum likelihood of a class [16,25]. The backpropagation is useful when the FCNs need to adjust its weights to improve the accuracy [16]. This is done using a cost optimizer and the best optimizer that has been used is the A**dam optimizer** as it is a combination of RMSProp and SGDNesterov, shown below in figure 13 [17].

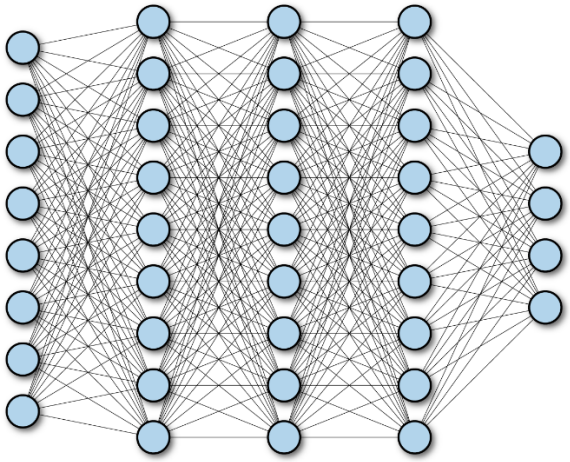


Figure 10: FCN to multi-class classification [27]

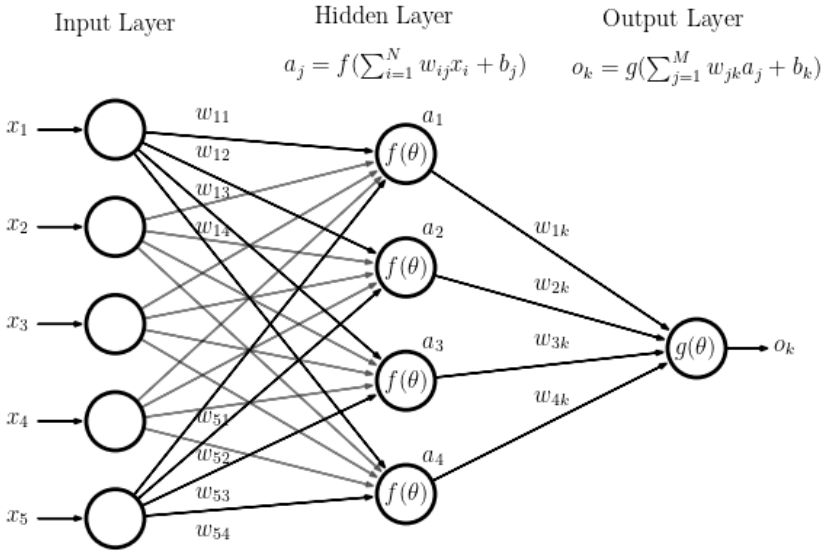


Figure 11: Three-layered Neural Network [19]

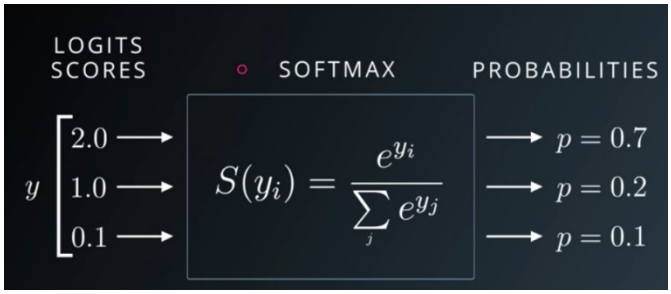


Figure 12: Softmax function used for the output [25]

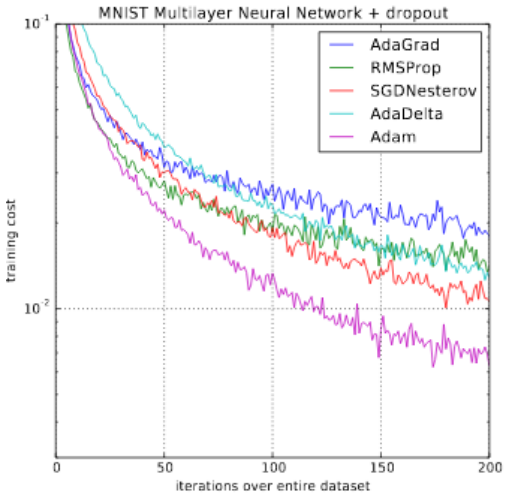


Figure 13: Cost optimizer comparison – Adam is shown to be the best [17]

Each convolutional layer holds the feature maps that represent the features such as edges, blobs etc. The FCNs hold the composite and aggregated information from all the convolutional layers that matter the most [15]. FCNs are useful for automatic feature learning just like convolutional layers. The only problem with FCNs compared to convolutional layers is that the number of parameters keeps increasing as we go from layer to layer in the ANN [21]. This is computationally heavy and so it is mainly used in the end of the convolutional architecture for prediction or classification. The main objective of the FCNs is to classify the images into a label. The output of the convolutional/pooling layers are reshaped/flattened into a single row of vector values shown in figure 14 below.

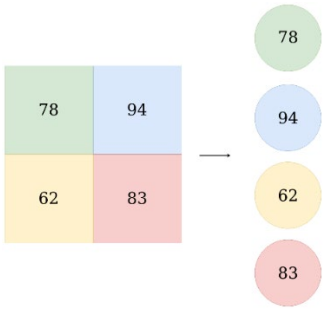


Figure 14: Unrolling or flattening of vector [20]

Transfer Learning

The pre-trained CNN this project will be using will be the VGG-16. VGG-16 participated in the ImageNet Large Scale Visual Recognition Challenge (ILSVRC) on 2014. Figure 15 shows the architecture of VGG-16. This architecture is the pre-trained network that this project will be using.

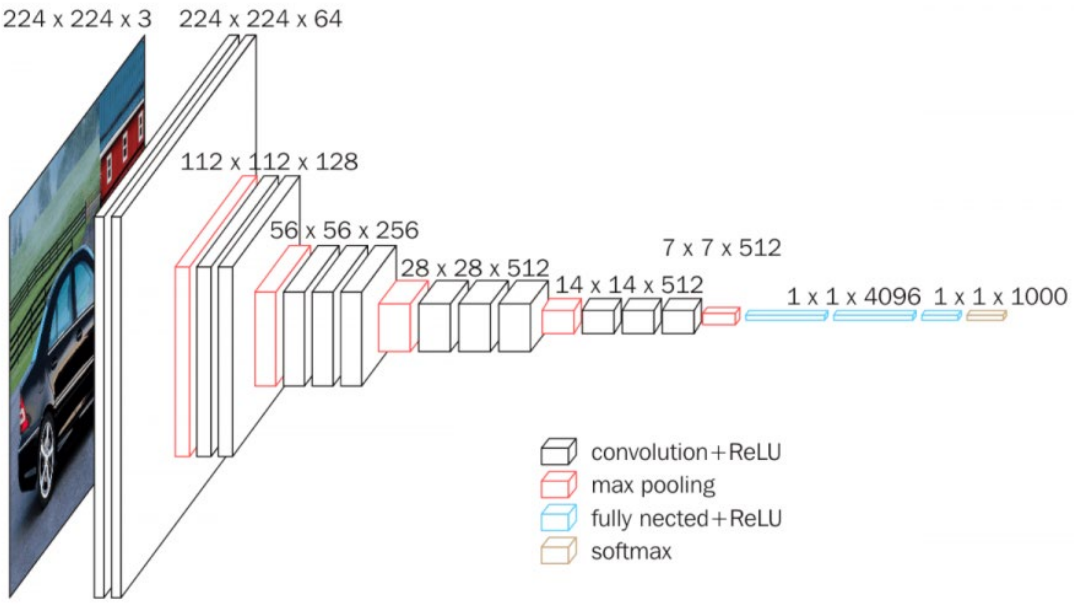


Figure 15: CNN architecture of VGG-16 [26]

In order to preprocess the data, to be used as input for the neural network, MATLAB was used. The other main task is to split the dataset into a training set, cross-validation set and a test set. The training set will be used to train the CNN with the inputs as labels. The cross-validation set will be used to modify any of the learning parameters such as the learning rate and mini batch size etc. Finally, the test set will be used for the CNN to make predictions on which pixel belongs to which class i.e. cancerous or not cancerous.

Python was used to implement the VGG-16 architecture for detection. The VGG-16 model takes in an input size of 96x96x3 for the images and then classifies which pixels in that

image are cancerous or not. In order to create the model architecture for semantic segmentation a U-net architecture shown in figure 16 below was implemented in VGG-16. This was done by using a down sampling and up sampling or in more specifically as max pooling and transposed convolutions.

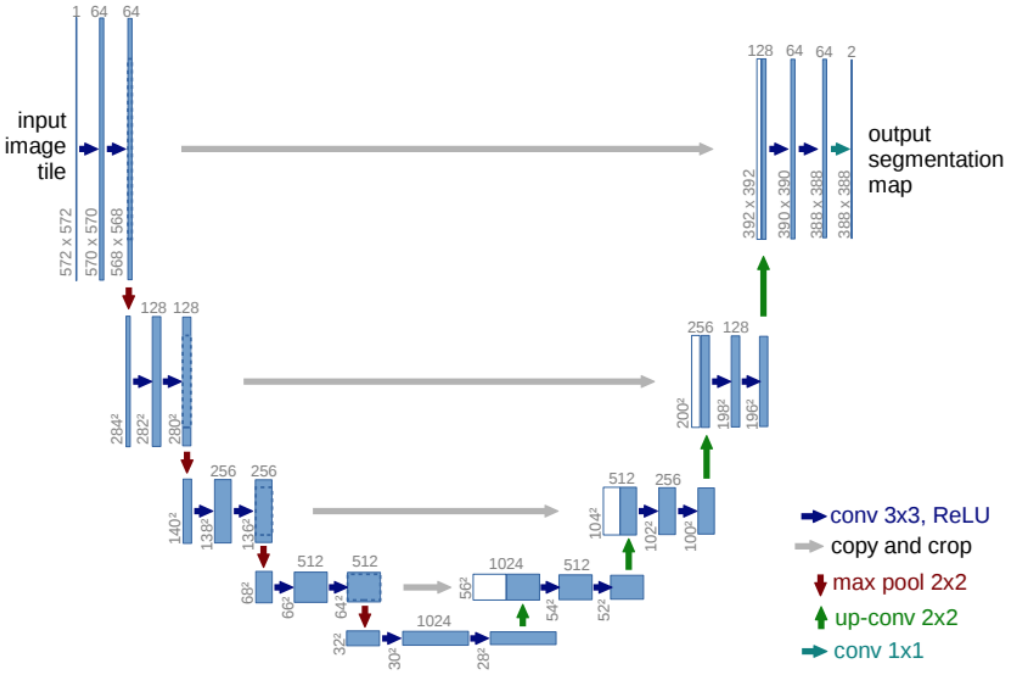


Figure 16: U-net architecture [28]

We down sample the features from the image by max pooling that reduces the size of the feature maps. The idea of max pooling as mentioned before was to retain only the important feature of each region. Down sampling basically converts a high-resolution image to a low-resolution image. It increases the number of channels/depth as the size of the image reduces but the receptive field increases [29].

The down sampling helps in identifying what the image is, which is useful for object detection. However, for semantic segmentation to work we need to know where are these mitotic

cells located in the image. This is the need to up sample the image i.e. convert from low-resolution to high-resolution using transposed convolution [29]. The problem lies when the up sampling and down sampling results in a bottleneck where important features cannot be transmitted to the decoder (up sampling layers). Therefore, we add skip connections as shown in the grey arrows in figure 16 above. These skip connections transfer the edges or blobs from the first convolutional layers to the up-sampling layers since some features may be lost in the up-sampling process.

CHAPTER III

RESULTS

The results we have got from the modifying the existing VGG-16 architecture is shown in the figure below. The accuracy and model loss is shown in figures 16 and 17 after running for 20 epochs. One epoch defines the time taken to complete one cycle of the full training dataset. The semantic segmentation where the pixels are classified as cancerous or mitotic are yellow and shown in figure 18.

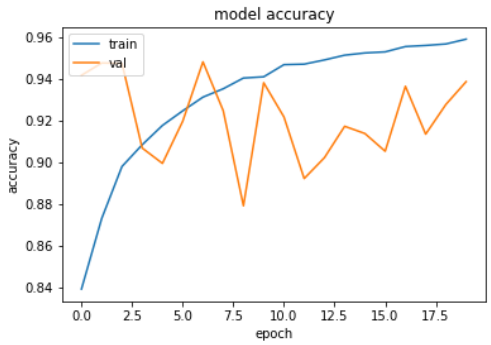


Figure 16: Model accuracy for 20 epochs

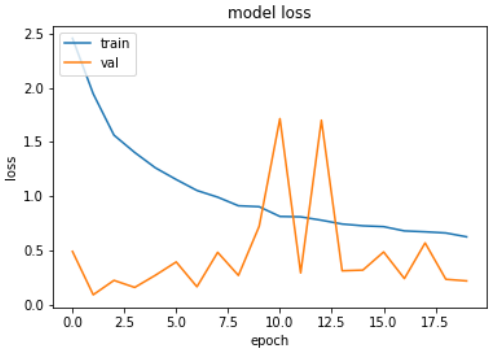


Figure 17: Model loss for 20 epochs

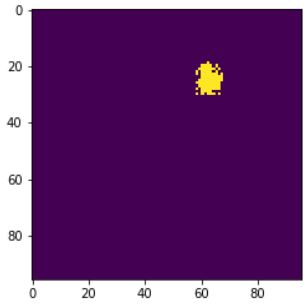


Figure 18(a)

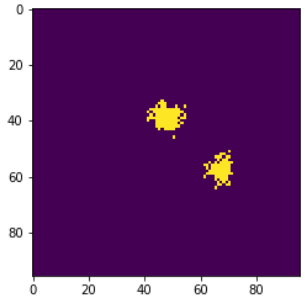


Figure 18(b)

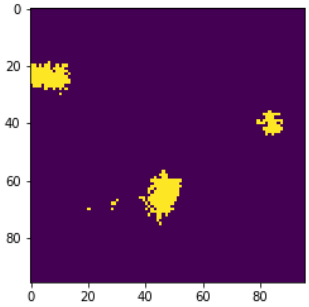


Figure 18(c)

Figure 18 a-c: Semantic segmentation results for some of the ICPR12 dataset

As we can see from figures 16 and 17 the validation accuracy and validation loss fluctuate quite rapidly. The purpose of the validation dataset is to generalize on new data. The model learns from the training data and it could memorize the locations of the mitotic cells in the image without being able to predict on new data. The problem of overfitting is when the model predicts

on all the patches of images that there are no cancerous cells and you would obtain really high accuracy for training and validation but when it comes to predicting the location of the mitotic cells it fails to recognize any mitotic cells. The accuracy and loss of the model of this overfitting problem is shown below in figures 19 and 20. The prediction or the output can be seen in figure 21 where all the pixels are of the same color. The model accuracy and loss do not reflect the 'true' outcome because the datasets contain an imbalance of mitotic cell and non-mitotic cell samples. However, there has been no extensive testing done and only a few samples of the test data have been shown here.

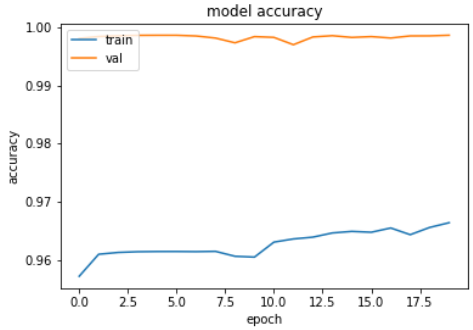


Figure 19: Model accuracy of overfitting case

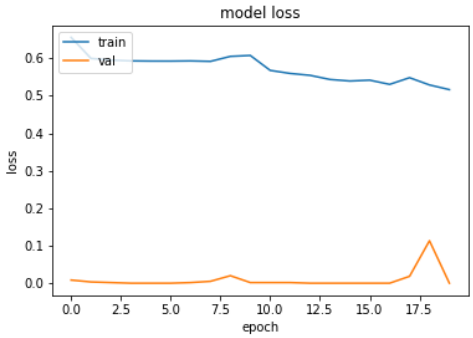


Figure 20: Model loss of overfitting case

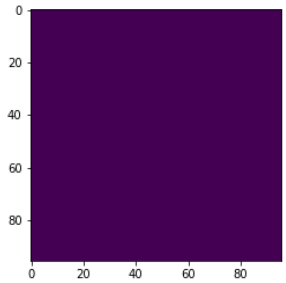


Figure 21: Results with no prediction of mitotic cells

CHAPTER IV

CONCLUSION

The convolutional neural network has shown great promise in detecting mitotic cells for breast cancer detection. Using transfer learning-based techniques have helped to improve the versatility of automatic cancer detection. Transfer learning has shown to be able to modify an existing architecture that is trained on the domain of classifying objects to classifying pixels on whether it is cancerous or not.

Although the model achieved a 96% on the training data the validation accuracy and loss will need to be improved. Factors such as overfitting by feeding the algorithm non-mitotic images and other reasons could influence the model to detect inaccurately. Because of imbalance between positive and negative samples in the datasets, the machine learning model accuracy does not reflect the mitotic cell detection accuracy. In the future, confusion matrix shall be used to provide more insights of the system performance.

This project is still ongoing and there will be some changes to the existing model such as with the preprocessing of the images, optimizer or even the architecture to achieve the goal of automating cancer detection. Unfortunately, the transfer learning to apply mitotic cell detection into animal histopathological images could not be completed because of time and resource limitation.

REFERENCES

[1] "imclose," *Types of Morphological Operations - MATLAB & Simulink*. [Online]. Available: https://www.mathworks.com/help/images/morphological-dilation-and-erosion.html . [Accessed: 09-Sep-2019].

[2] V. Badrinarayanan, A. Kendall, and R. Cipolla, "Segnet: A deep convolutional encoder-decoder architecture for image segmentation," 10-Oct-2016. Available: https://arxiv.org/pdf/1511.00561.pdf.

[3] L. von Chamier, R. F. Laine, and R. Henriques, "Artificial intelligence for microscopy: what you should know," *Biochemical Society transactions*, 30-Aug-2019. Available: https://www.ncbi.nlm.nih.gov/pubmed/31366471.

[4] G. Jiménez and D. Racoceanu, "Deep Learning for Semantic Segmentation vs. Classification in Computational Pathology: Application to Mitosis Analysis in Breast Cancer Grading," *Frontiers*, 29-May-2019. Available: https://www.frontiersin.org/articles/10.3389/fbioe.2019.00145/full .

[5] H. Chen, Q. Dou, X. Wang, J. Qin, and P.-A. Heng, "Mitosis detection in breast cancer histology images via deep cascaded networks," *ACM Digital Library*, 2016. Available: https://dl.acm.org/citation.cfm?id=3015984.

[6] "Dataset," MICCAI Grand Challenge: Tumor Proliferation Assessment Challenge (TUPAC16), 24-Aug-2016. [Online]. Available: http://tupac.tue-image.nl/node/3 . [Accessed: 13-Sep-2019].

[7] M. Veta, P. Diest, S. Willems, H. Wang, A. Madabhushi, A. Cruz-Roa, F. Gonzalez, A. Larsen, J. Vestergaard, A. Dahl, D. Ciresan, J. Schmidhuber, A. Giusti, L. Gambardella, F. Tek, T. Walter, C.-W. Wang, S. Kondo, B. Matuszewski, F. Precioso, V. Snell, J. Kittler, T. Campos, A. Khan, N. Rajpoot, E. Arkoumani, M. Lacle, M. Viergever, and J. Pluim, "Assessment of algorithms for mitosis detection in breast cancer histopathology images," *arXiv.org*, 21-Nov-2014. [Online]. Available: https://arxiv.org/abs/1411.5825 . [Accessed: 13-Sep-2019].

[8] "MITOS-ATYPIA-14 - Grand Challenge," *GrandChallenge*. [Online]. Available: https://mitos-atypia-14.grand-challenge.org/ . [Accessed: 13-Sep-2019].

[9] “Breast cancer,” World Health Organization, 12-Sep-2018. [Online]. Available: https://www.who.int/cancer/prevention/diagnosis-screening/breast-cancer/en/ . [Accessed: 14-Sep-2019].

[10] “U.S. Breast Cancer Statistics,” Breastcancer.org, 13-Feb-2019. [Online]. Available: https://www.breastcancer.org/symptoms/understand_bc/statistics

[11] H. Su, F. Liu, Y. Xie, F. Xing, S. Meyyappan, and L. Yang, “Region segmentation in histopathological breast cancer images using deep convolutional neural network: Semantic Scholar,” 01-Jan-1970. [Online]. Available: https://www.semanticscholar.org/paper/Region-segmentation-in-histopathological-breast-Su-Liu/31cf1c5198a37a8bee0e9a497744a110dcdb72f1 [Accessed: 23-Feb-2020].

[12] K. Chandrasekhar, R. P. Kumar, P. Bharathi, and K. V. Triveni, “HISTOPATHOLOGICAL CANCER DETECTION,” International Research Journal of Computer Science (IRJCS), vol. 6, no. 04, pp. 102–124, Apr. 2019.

[13] A. Amini, *Convolutional Neural Networks | MIT 6.S191*. 2020 [Online]. Available: https://www.youtube.com/watch?v=iaSUYvmCekI&t=240s. [Accessed: 01- Mar- 2020]

[14] "Convolutional Neural Network", *Medium*, 2020. [Online]. Available: https://towardsdatascience.com/convolutional-neural-network-17fb77e76c05. [Accessed: 02- Mar- 2020]

[15] "What happens in the fully connected layer in a convolutional neural network?", *quora*, 2020. [Online]. Available: https://www.quora.com/What-happens-in-the-fully-connected-layer-in-a-convolutional-neural-network. [Accessed: 01- Mar- 2020]

[16] "Fully Connected Layers in Convolutional Neural Networks: The Complete Guide - MissingLink.ai", *MissingLink.ai*, 2020. [Online]. Available: https://missinglink.ai/guides/convolutional-neural-networks/fully-connected-layers-convolutional-neural-networks-complete-guide/. [Accessed: 02- Mar- 2020]

[17] V. Bushaev, "Adam—latest trends in deep learning optimization.", *Medium*, 2020. [Online]. Available: https://towardsdatascience.com/adam-latest-trends-in-deep-learning-optimization-6be9a291375c. [Accessed: 02- Mar- 2020]

[18] A. Kizrak, "Comparison of Activation Functions for Deep Neural Networks", *Medium*, 2020. [Online]. Available: https://towardsdatascience.com/comparison-of-activation-functions-for-deep-neural-networks-706ac4284c8a. [Accessed: 02- Mar- 2020]

[19] "Neural Network Diagram — astroML 0.4 documentation", *Astroml.org*, 2020. [Online]. Available: https://www.astroml.org/book_figures/chapter9/fig_neural_network.html. [Accessed: 02- Mar- 2020]

[20] Ashushekar, "ashushekar/image-convolution-from-scratch," *GitHub*, 16-Nov-2019. [Online]. Available: https://github.com/ashushekar/image-convolution-from-scratch. [Accessed: 23-Feb-2020].

[21] A. Gad, "Derivation of Convolutional Neural Network from Fully Connected Network Step-By-Step", *Medium*, 2020. [Online]. Available: https://towardsdatascience.com/derivation-of-convolutional-neural-network-from-fully-connected-network-step-by-step-b42ebafa5275. [Accessed: 02- Mar- 2020]

[22] N. Ketkar, *Deep Learning with Python - A hands-on introduction*. Bangalore, India: Apress, 2017, pp. 27-28, 61-76.

[23] Edureka, *Convolutional Neural Network (CNN) | Convolutional Neural Networks With TensorFlow | Edureka*. 2020 [Online]. Available: https://www.youtube.com/watch?v=umGJ30-15_A. [Accessed: 10- Feb- 2020]

[24] "What is translation invariance in computer vision and convolutional neural network?", *Cross Validated*, 2020. [Online]. Available: https://stats.stackexchange.com/questions/208936/what-is-translation-invariance-in-computer-vision-and-convolutional-neural-netwo . [Accessed: 01- Mar- 2020]

[25] Uniqtech, "Understand the Softmax Function in Minutes", *Medium*, 2020. [Online]. Available: https://medium.com/data-science-bootcamp/understand-the-softmax-function-in-minutes-f3a59641e86d. [Accessed: 02- Mar- 2020]

[26] “VGG16 - Convolutional Network for Classification and Detection,” *VGG16 - Convolutional Network for Classification and Detection*, 21-Nov-2018. [Online]. Available: https://neurohive.io/en/popular-networks/vgg16/. [Accessed: 23-Feb-2020].

[27] "TensorFlow for Deep Learning", *O'Reilly Online Learning*, 2020. [Online]. Available: https://www.oreilly.com/library/view/tensorflow-for-deep/9781491980446/ch04.html. [Accessed: 11- Mar- 2020]

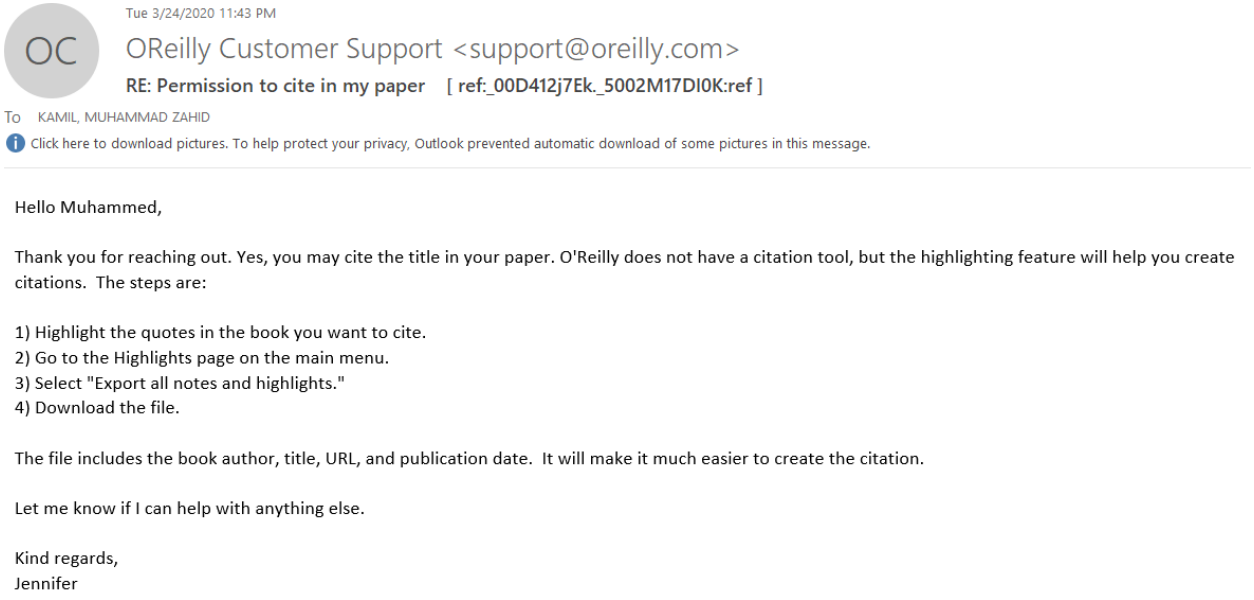
[28] C. Shorten, "U-Net: Convolutional Networks for Biomedical Image Segmentation", *Medium*, 2018. [Online]. Available: https://medium.com/@connorshorten300/u-net-convolutional-networks-for-biomedical-image-segmentation-c9140032f92. [Accessed: 01- Apr- 2020]

[29] H. Lamba, "Understanding Semantic Segmentation with UNET", *Medium*, 2019. [Online]. Available: https://towardsdatascience.com/understanding-semantic-segmentation-with-unet-6be4f42d4b47. [Accessed: 01- Apr- 2020]

APPENDIX

Statement of Permission

Permission for reference [27]



Tue 3/24/2020 11:43 PM

OC

OReilly Customer Support <support@oreilly.com>

RE: Permission to cite in my paper [ref:_00D412j7Ek._5002M17DI0K:ref]

To KAMIL, MUHAMMAD ZAHID

Click here to download pictures. To help protect your privacy, Outlook prevented automatic download of some pictures in this message.

Hello Muhammed,

Thank you for reaching out. Yes, you may cite the title in your paper. O'Reilly does not have a citation tool, but the highlighting feature will help you create citations. The steps are:

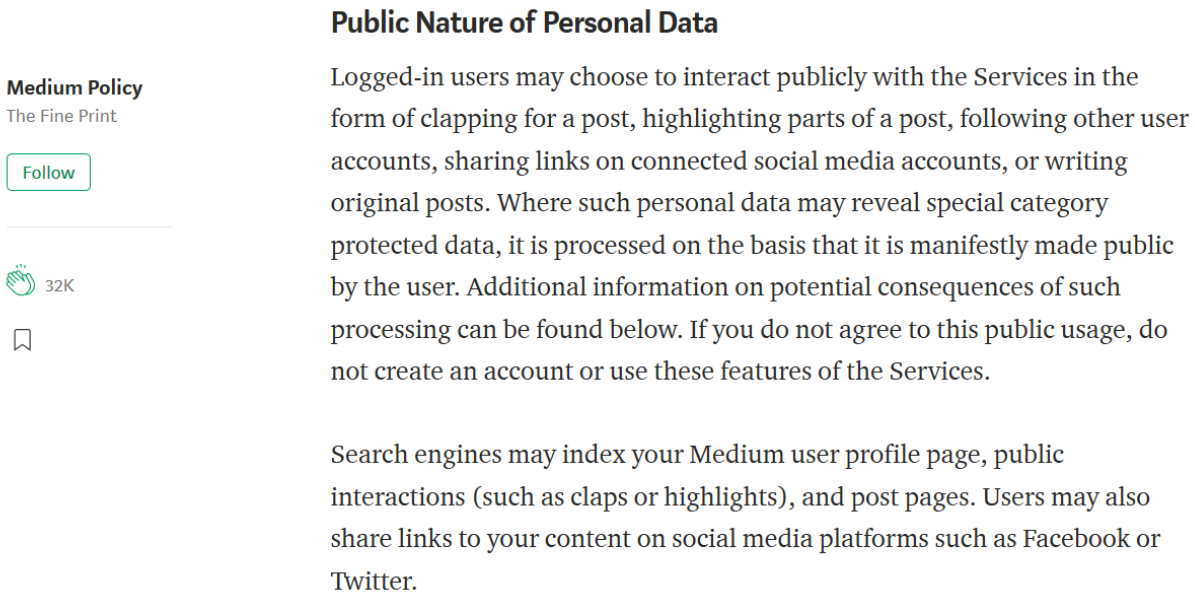
1) Highlight the quotes in the book you want to cite.
2) Go to the Highlights page on the main menu.
3) Select "Export all notes and highlights."
4) Download the file.

The file includes the book author, title, URL, and publication date. It will make it much easier to create the citation.

Let me know if I can help with anything else.

Kind regards,
Jennifer

Permission for references [18,25,26,28]



Medium Policy

The Fine Print

Follow

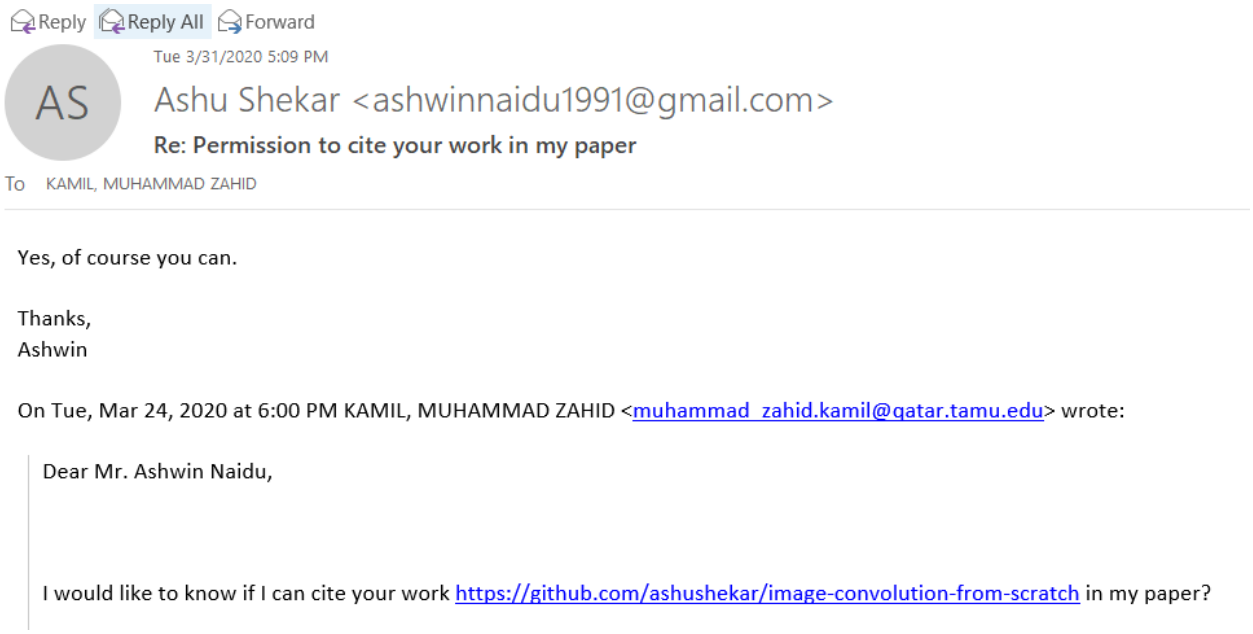
32K

Public Nature of Personal Data

Logged-in users may choose to interact publicly with the Services in the form of clapping for a post, highlighting parts of a post, following other user accounts, sharing links on connected social media accounts, or writing original posts. Where such personal data may reveal special category protected data, it is processed on the basis that it is manifestly made public by the user. Additional information on potential consequences of such processing can be found below. If you do not agree to this public usage, do not create an account or use these features of the Services.

Search engines may index your Medium user profile page, public interactions (such as claps or highlights), and post pages. Users may also share links to your content on social media platforms such as Facebook or Twitter.

Permission for references [20]



Reply Reply All Forward

Tue 3/31/2020 5:09 PM

AS

Ashu Shekar <ashwinnaidu1991@gmail.com>

Re: Permission to cite your work in my paper

To KAMIL, MUHAMMAD ZAHID

Yes, of course you can.

Thanks,
Ashwin

On Tue, Mar 24, 2020 at 6:00 PM KAMIL, MUHAMMAD ZAHID <muhammad_zahid.kamil@qatar.tamu.edu> wrote:

> Dear Mr. Ashwin Naidu,
>
> I would like to know if I can cite your work https://github.com/ashushekar/image-convolution-from-scratch in my paper?