Machine learning techniques for galaxy imagery and photometry

by

Hunter Goddard

B.S., Kansas State University, 2017

M.S., Kansas State University, 2021

AN ABSTRACT OF A DISSERTATION

submitted in partial fulfillment of the
requirements for the degree

DOCTOR OF PHILOSOPHY

Department of Computer Science
Carl R. Ice College of Engineering

KANSAS STATE UNIVERSITY
Manhattan, Kansas

2023

Abstract

In the past two decades, autonomous digital sky surveys have enabled significant advances in astronomy by collecting massive databases of imagery and other information. The quantity of data, coupled with the variety of scientific questions that require its analysis, makes manual analysis of these data impractical. To address this challenge, machine learning algorithms have been widely adopted for data analysis and product generation in astronomy. In this dissertation I examine the efficacy of machine learning algorithms such as deep convolutional neural networks, support vector machines, and vision transformers for the purpose of astronomical data analysis, with emphasize on extra-galactic objects. These include algorithms that can annotate large datasets of galaxy images, and their application to premier digital sky surveys such as Pan-STARRS. Specifically, I address the following research question: How effective are machine learning algorithms for annotating astronomical data, and what are the downsides of using these algorithms for this purpose? Namely, biases that are typical to machine learning systems can influence the annotations, which may consequently lead to false conclusions when applying statistical analysis to data annotated using such systems. These biases are often difficult to identify. Overall, this research highlights the importance of careful consideration of machine learning algorithms and their potential biases when applying them to astronomical data analysis. Our findings have broad implications for the use of machine learning in astronomy and other scientific domains, as they demonstrate the importance of addressing potential biases in machine learning systems to avoid erroneous scientific conclusions.

Machine learning techniques for galaxy imagery and photometry

by

Hunter Goddard

B.S., Kansas State University, 2017

M.S., Kansas State University, 2021

A DISSERTATION

submitted in partial fulfillment of the
requirements for the degree

DOCTOR OF PHILOSOPHY

Department of Computer Science
Carl R. Ice College of Engineering

KANSAS STATE UNIVERSITY
Manhattan, Kansas

2023

Approved by:

Major Professor
Lior Shamir

Copyright

Abstract

In the past two decades, autonomous digital sky surveys have enabled significant advances in astronomy by collecting massive databases of imagery and other information. The quantity of data, coupled with the variety of scientific questions that require its analysis, makes manual analysis of these data impractical. To address this challenge, machine learning algorithms have been widely adopted for data analysis and product generation in astronomy. In this dissertation I examine the efficacy of machine learning algorithms such as deep convolutional neural networks, support vector machines, and vision transformers for the purpose of astronomical data analysis, with emphasize on extra-galactic objects. These include algorithms that can annotate large datasets of galaxy images, and their application to premier digital sky surveys such as Pan-STARRS. Specifically, I address the following research question: How effective are machine learning algorithms for annotating astronomical data, and what are the downsides of using these algorithms for this purpose? Namely, biases that are typical to machine learning systems can influence the annotations, which may consequently lead to false conclusions when applying statistical analysis to data annotated using such systems. These biases are often difficult to identify. Overall, this research highlights the importance of careful consideration of machine learning algorithms and their potential biases when applying them to astronomical data analysis. Our findings have broad implications for the use of machine learning in astronomy and other scientific domains, as they demonstrate the importance of addressing potential biases in machine learning systems to avoid erroneous scientific conclusions.

Table of Contents

List of Figures ix

List of Tables xi

Acknowledgements xiii

1 A Catalog of Broad Morphology of Pan-STARRS Galaxies Based on Deep Learning 1

1.1 Introduction 1

1.2 Data 3

1.3 Image analysis method 4

1.3.1 Primary classification 6

1.3.2 Secondary Classification 7

1.4 Results 8

1.4.1 Comparison to existing SDSS catalog 9

1.5 Conclusions 11

2 SVMnet: Non-parametric Image Classification Based on Convolutional Ensembles of Support Vector Machines for Small Training Sets 15

2.1 Introduction 15

2.2 Architecture of SVMnet 18

2.2.1 Dropout 21

2.2.2 Formal definition of SVMnet 21

2.3 Experimental Results 23

2.3.1 COIL-100 Object Recognition 24

2.3.2 Imagenette 24
2.3.3 COVID-19 Radiography 25
2.3.4 Astronomical image data 27
2.3.5 WND-CHARM 28
2.3.6 Computational complexity 28
2.3.7 Architecture Comparison 31
2.4 Conclusions 32

3 Machine Learning and Neural Network Bias in Analysis of Photometric Data . . . 35
3.1 Introduction 35
3.2 Data 37
3.3 Methodology 39
3.4 Generalizing to Different Areas of the Sky 41
3.4.1 Annotation of the SDSS with training sets from different sky regions 43
3.4.2 Pan-STARRS by Sky Region 45
3.5 Generalizing to Different Sky Surveys 48
3.5.1 Annotating the SDSS and Pan-STARRS 50
3.6 Conclusions 51

4 Experiments with Vision Transformers for Fine-Grained Galaxy Morphology Classification 54
4.1 Introduction 54
4.2 Galaxy Zoo 55
4.3 Experiment Design 57
4.3.1 Vision Transformer 60
4.3.2 Training Procedure 62
4.4 Experimental Results 64
4.5 Conclusions 67

Bibliography 69

List of Figures

1.1 Distribution of the r Kron magnitude of the galaxies in the dataset. 3

1.2 Number of spiral and elliptical galaxies remaining when keeping only those at or above a certain model confidence. 9

1.3 The proportion of predicted labels that, when restricted to a minimum confidence threshold, agree with the annotations in KS16. For example, restricting our catalog to labels with 90% confidence or higher results in approximately 98% agreement between the catalogs. 11

2.1 Example of a simple weighted layer of SVMnet. Each node in the layer is an SVM, trained with a subset of the inputs (pixels). Weight outputs are optional for a given layer. 19

2.2 Example SVMnet architecture containing two SVM layers (in green) and a class label vote. Each SVM is trained on a patch of the layer's input. An $n \times m$ SVM layer produces $n \times m \times d$ output ($d \geq 1$). 20

2.3 Test-set accuracy (left) and training time (right) of SVMnet and ResNet on COIL-100 images when fitted with different training set sizes. 25

2.4 Test-set accuracy (left) and training time (right) of SVMnet and ResNet on Imagenette when fitted with different training set sizes. 26

2.5 Test-set accuracy (left) and training time (left) of SVMnet and ResNet on COVID-19 chest x-ray images when fitted with different training set sizes. The accuracy of the ResNet models displays considerable overlap. 27

2.6 Test-set accuracy (left) and training time (right) of SVMnet and ResNet on Pan-STARRS galaxy images when fitted with different training set sizes. . . 28

2.7 Prediction accuracy of one-layer SVMnet architectures fitted to COIL-100. Each group of three box plots represents the same window size with stride length 3, 5, and 7, respectively. Each box plot shows the distribution in model accuracy when using five training sets of 200-1000 examples. 31

2.8 Training times for one-layer SVMnet architectures fitted to COIL-100. 32

3.1 Histogram of SDSS galaxy magnitudes by morphological class. 39

3.2 Histogram of Pan-STARRS galaxy magnitudes by morphological class. . . . 40

3.3 Training and validation loss of the neural network architecture with the best performance for the SDSS. 44

3.4 Average test-set accuracy of the SDSS model with different training set sizes. 44

3.5 Training and validation loss of the winning neural network architecture for Pan-STARRS. 46

3.6 Average test-set accuracy of the Pan-STARRS model with different training set sizes. 47

3.7 Training and validation loss of the neural network architecture with the best performance when trained with a mixture of data from the SDSS and Pan-STARRS. 50

4.1 Training and validation loss of our ViT on the combined Galaxy Zoo dataset. 65

4.2 Training and validation loss of the ViT when trained on each Galaxy Zoo dataset individually. 65

List of Tables

1.1 Examples of images filtered for having too many saturated pixels or having too few foreground pixels. 5

1.2 Galaxies imaged by both Pan-STARRS and HST. While the Pan-STARRS images do not show clear spiral arms of the galaxies, HST shows that these galaxies are clearly spiral, and the arms can be identified. 10

1.3 Examples of images that were misclassified by the model. 12

1.4 Examples of images that were classified correctly by the model. 13

2.1 Comparison of the classification accuracy of WND-CHARM and SVMnet when trained on a small number of samples from four datasets. 29

2.2 Comparison of the response time (in seconds) of SVMnet and ResNet to predict class labels for 1, 10, 100, and 1000 samples of the COIL-100 dataset. . 30

3.1 Feature vector summary for SDSS and Pan-STARRS datasets. 38

3.2 The size and position of the constellations selected for analysis. Right ascension (RA) and declination (Dec) are given as midpoints of the constellation boundaries. 42

3.3 The number of samples for each type of galaxy within each sky region available in either dataset. 42

3.4 Confusion matrices of Hercules galaxy morphology predictions from two models with training data from different regions. Rows represent true labels and columns the predicted labels. 43

3.5 Prediction accuracy of SDSS models by training set region and test set region. 45

3.6 SDSS binomial test p-values. Statistical significance ($p < 0.05$) is indicated in bold. 46

3.7 Prediction accuracy of Pan-STARRS models by training set region and test set region. 47

3.8 Pan-STARRS binomial test p-values. Statistical significance ($p < 0.05$) is indicated in bold. 48

3.9 Unsupervised clustering using a 2-component Gaussian mixture model. . . . 49

3.10 Test-set classification accuracy by models trained on different combinations of data from the SDSS and Pan-STARRS (PS1). 51

4.1 The number of labeled images used from each Galaxy Zoo project. 56

4.2 A list of tasks and responses in the four Galaxy Zoo projects used in the experiment. In practice, each task included example diagrams to assist the user in choosing their response. 58

4.3 A list of tasks and responses in the four Galaxy Zoo projects used in the experiment (continued). 59

4.4 Example images from each Galaxy Zoo project with the validation crop and examples of possible augmentations during the training step. 63

4.5 The accuracy of the combined model on each project's subset of the combined test set, and the accuracy of each individually-trained model on its respective test set. 66

Acknowledgments

This research was funded by NSF grant AST-1903823. The Pan-STARRS1 Surveys (PS1) and the PS1 public science archive have been made possible through contributions by the Institute for Astronomy, the University of Hawaii, the Pan-STARRS Project Office, the Max-Planck Society and its participating institutes, the Max Planck Institute for Astronomy, Heidelberg and the Max Planck Institute for Extraterrestrial Physics, Garching, The Johns Hopkins University, Durham University, the University of Edinburgh, the Queen's University Belfast, the Harvard-Smithsonian Center for Astrophysics, the Las Cumbres Observatory Global Telescope Network Incorporated, the National Central University of Taiwan, the Space Telescope Science Institute, the National Aeronautics and Space Administration under Grant No. NNX08AR22G issued through the Planetary Science Division of the NASA Science Mission Directorate, the National Science Foundation Grant No. AST-1238877, the University of Maryland, Eotvos Lorand University (ELTE), the Los Alamos National Laboratory, and the Gordon and Betty Moore Foundation.

Chapter 1

A Catalog of Broad Morphology of Pan-STARRS Galaxies Based on Deep Learning

This chapter was published in Goddard, H., Shamir, L., A catalog of broad morphology of Pan-STARRS galaxies based on deep learning, *Astrophysical Journal Supplement Series*, 251(2), 28. IOP, 2020

1.1 Introduction

With their ability to generate very large databases, autonomous digital sky surveys have been enabling research tasks that were not possible in the pre-information era, and have been becoming increasingly pivotal in astronomy. The ability of digital sky surveys to image large parts of the sky, combined with the concept of virtual observatories that make these data publicly accessible[1], has been introducing a new form of astronomy research, and that trend is bound to continue[2;3].

The Panoramic Survey Telescope and Rapid Response System (Pan-STARRS)[4;5] is a comprehensive digital sky survey covering $\sim 10^3$ degree^2 per night using an array of two

1.8m telescopes. Among other celestial objects, Pan-STARRS images a very large number of galaxies. Due to the complexity of galaxy morphology, the ability of current photometric pipelines to analyze these galaxy images is limited, and substantial information that is visible to the humans eye is practically unavailable to users of digital sky surveys data.

To automate the analysis of galaxy images, several methods have been proposed, including GALFIT[6], GIM2D[7], CAS[8], the Gini coefficient of the light distribution[9], Ganalyzer[10], and SpArcFiRe[11]. However, the ability of these methods to analyze a large number of real-world galaxy images and produce clean data products is limited, and catalogs of galaxy morphology were prepared manually by professional astronomers[12;13].

Due to the high volumes of data, the available pool of professional astronomers is not able to provide the sufficient labor to analyze databases generated by modern digital sky surveys, leading to the use of crowdsourcing for that task[14–16]. The main crowdsourcing campaign for analysis of galaxy morphology was Galaxy Zoo[15], providing annotations of the broad morphology of galaxies imaged by Sloan Digital Sky Survey (SDSS), as well as other surveys such as the Cosmic Assembly Near-infrared Deep Extragalactic Legacy (CANDELS). However, analyzing the broad morphology of SDSS galaxies required $\sim$3 years of work performed by over 10^5 volunteers, and led to $\sim 7 \cdot 10^4$ galaxies considered "superclean". Given the huge databases of current and future sky surveys, it is clear that even when using crowdsourcing, the throughout of manual classification might not be sufficient for an exhaustive analysis of such databases.

The use of machine learning provided more effective methods for the purpose of galaxy image classification[17–26], and the use of such methods also provided computer-generated catalogs of galaxy morphology[27–35]. Automatic annotation methods were also tested on Pan-STARRS data by using the photometric features of colors and moments, classified by a Random Forest classifier[36].

Here we use automatic image analysis to prepare a catalog of the broad morphology of $\sim 1.7 \cdot 10^6$ Pan-STARRS DR1 galaxies. The catalog was generated by using a data analysis process that involves several steps and two convolutional neural networks (CNN) that automated the annotation process to handle the high volume of data.

1.2 Data

The galaxy image data is sourced from the first data release (DR1) of Pan-STARRS[5;37;38]. First, all objects with Kron r magnitude of less than than 19 and identified by Pan-STARRS photometric pipeline as extended in all bands were selected.

To filter objects that are too small to identify morphology, objects that have Petrosian radius smaller than 5.5" were removed. To remove stars, objects that their PSF i magnitude subtracted by their Kron i magnitude was greater than 0.05 were also removed. That led to a dataset of 2,394,452 objects[33]. Objects that were flagged by Pan-STARRS photometric pipeline as artifacts, had a brighter neighbor, defect, double PSF, or a blend in any of the bands were excluded from the dataset. That led to a dataset of 2,131,371 objects assumed to be sufficiently large and clean to allow morphological analysis. Figure 1.1 shows the distribution of the r Kron magnitude of the galaxies in the dataset.

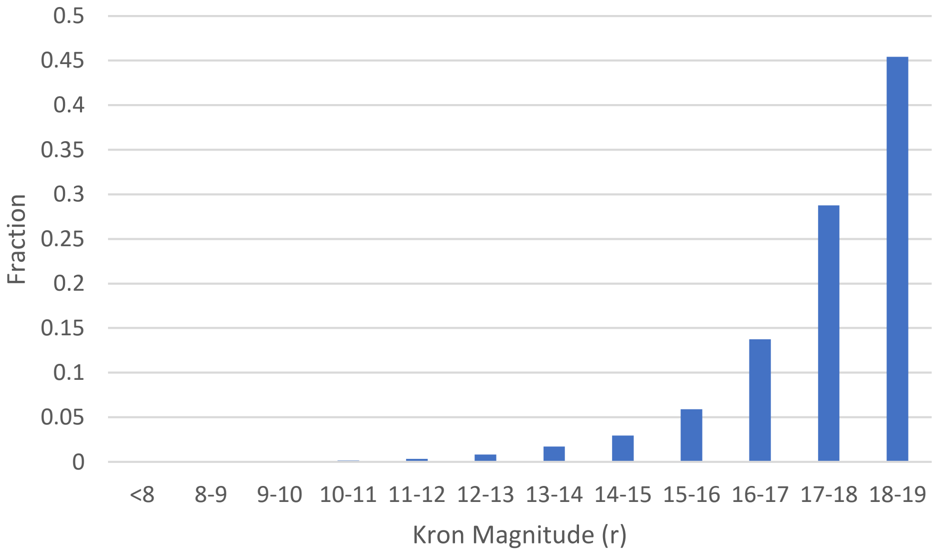


Figure 1.1: Distribution of the r Kron magnitude of the galaxies in the dataset.

The galaxy images were then downloaded using Pan-STARRS *cutout* service. The images are in the JPG format and have a dimensionality of 120×120 pixels. Pan-STARRS *cutout* provides JPG images for each of the bands. Here we use the images of the g band, as the color images using the y, i, and g bands are in many cases noisy, and do not allow effective analysis of the morphology. The process of downloading the data was completed in 62 days.

The initial scale of the *cutout* was set to 0.25” per pixel. For each image that was downloaded, all bright pixels (grayscale value higher than 125) located on the edge of the frame were counted. If more than 25% of the pixels on the edge of the frame were bright, it is an indication that the object does not fully fit inside the frame. In that case, the scale was increased by 0.05”, and the image was downloaded again. That was repeated until the number of bright pixels on the edge was less than 25% of the total edge pixels, meaning that the object is inside the frame. The JPG images are far smaller than the FITS images. A 120×120 JPG image retrieved through Pan-STARRS *cutout* service is normally of size of ∼3KB, while an image of the same size in the FITS format will be ∼76KB. Although the FITS files provide more information, downloading the files in FITS format requires more time, and does not fit the very large number of files used for the purpose of providing a catalog. The JPG images do not allow photometry, but they are smaller than the FITS files and provide information about the shape of the galaxy, which is the information required for the morphological classification of the galaxies. As explained in Section 1.3.1, the training of the neural network was done with images retrieved from Pan-STARRS, with the exact same size and format as the images that were annotated.

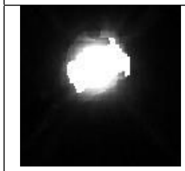
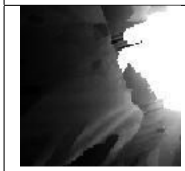
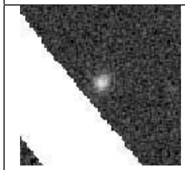
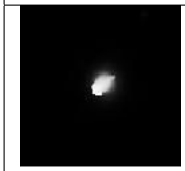
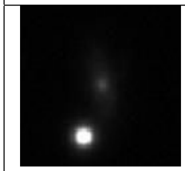
1.3 Image analysis method

The filtering of the data described in Section 1.2 aims at removing objects that are not clean galaxy images. That allows to reduce the number of images downloaded and classified in the next step with the deep neural network. The removal of objects that are not galaxy images also makes the neural network more accurate due to the higher consistency of the data it is trained with.

To remove saturated images and images that have too few features to allow morphological classification, two additional filters are used. The first filter finds the ratio of fully saturated pixels (a grayscale value of 255 in the JPG image) to the total number of pixels and discards the image if this ratio is higher than 15:1000. Since a high number of saturated pixels is not expected in a clean galaxy image, the simple threshold of 1.5% is sufficient to identify and reject saturated images that are not galaxy images. This step rejected 30,220 objects that were identified as saturated.

The second filter uses the Otsu global threshold method[39] to separate the image into foreground and background pixels. If the number of foreground pixels is less than 1.8% of the total image, the image is marked as having too few distinguishable features. This filter rejected 375,107 galaxies that were identified as having too little foreground to allow identification. Together, these filters removed 405,327 images (∼19%) from the data set. The thresholds were determined experimentally by observing galaxy image samples. Table 1.1 shows examples of several objects that were filtered based on too few foreground pixels or too many saturated pixels.

Table 1.1: Examples of images filtered for having too many saturated pixels or having too few foreground pixels.

Image	Saturated pixels (%)	Foreground pixels (%)
	6.1	10.7
	13.5	21.7
	30.9	34.9
	0.06	1.4
	0.16	1.1

1.3.1 Primary classification

The classifier used for the purpose of annotating the galaxy images is a deep convolutional neural network (DCNN) based on the LeNet-5 architecture[40]. To adjust the model for input images of size 120×120 instead of 32×32, the kernel in the first convolutional layer was changed from 5×5 with stride 1 to 10×10 with stride 2, and the filter in the first pooling layer was similarly changed from 2×2 with stride 2 to 4×4 with stride 4. Each of the following layers has identical hyperparameters except for the output layer, where the number of classes is reduced from 10 to 2. The SoftMax output layer of the model provides a degree of certainty for the annotations that allows controlling the size/accuracy trade-off of the catalog, as will be discussed in Section 1.4.

Training samples were obtained using the debiased "superclean" Galaxy Zoo annotations. "Superclean" objects are objects on which 95% or more of the annotators agreed on their morphology with correction for the redshift bias[15]. That selection leads to a subset of very consistent annotations[15], but it also filters the vast majority of Galaxy Zoo objects that do not satisfy these requirements. The Galaxy Zoo crowdsourcing campaign annotated SDSS galaxies, which are imaged with a different instrument and image processing pipeline. Since consistency between the training and test data is important in machine learning systems, galaxies imaged by the SDSS cannot be safely used to train an artificial neural network that classifies galaxies imaged by Pan-STARRS. To be able to classify Pan-STARRS images effectively, the neural network needs to be trained with galaxies imaged by Pan-STARRS, and processed through the same image processing pipeline. Although it has been shown that neural networks trained with data from one telescope can be used to classify data from other telescopes[41], it has also been shown that the accuracy of such networks is inferior to the accuracy of neural network trained and tested with data from the same instrument[41]. In order to train the neural network with images from the same dataset it is expected to annotate, the images of the galaxies annotated by Galaxy Zoo were retrieved from Pan-STARRS. Pan-STARRS has a different footprint than the SDSS, so not all galaxies annotated by Galaxy Zoo are also imaged by Pan-STARRS. However, 22,456 Galaxy Zoo galaxies with "superclean" annotations were matched with galaxies in Pan-STARRS DR1 based on their right ascension and declination (within difference of 0.0001 degrees). These images were fetched from Pan-STARRS and were used for training the neural network.

Galaxy Zoo manual annotations have been shown in the past to be sensitive to the spin direction of the galaxies[42]. To eliminate the possible effect of spin patterns, the training set was augmented such that all galaxies were mirrored (i.e. reflected across the vertical axis), and both the original and mirrored image of each galaxy were used in the training set. That resulted in a training set of 31,564 spiral images and 13,348 elliptical images. Mirroring the spiral galaxies ensures a symmetric dataset that is not biased by certain preferences of the human volunteers who annotated the galaxies. That is, while mirroring the images in the training set is often used when training deep neural networks for augmenting the data and increasing the number of training samples, in this case it was also used to produce a symmetric unbiased dataset. Mirroring of the elliptical galaxies was done to ensure consistency in how the training data are handled, and avoid a situation in which different classes are handled differently.

The classifier is implemented in Python 3 using TensorFlow and Keras. The model was trained for 250 epochs on a 70% training subset and ended with 96% accuracy when evaluated against the remaining 30% testing subset. Loss was computed using categorical cross entropy, and stochastic gradient descent (SGD) was used as the optimizer. Various activation functions including ReLU were tested with preliminary data and gave comparable classification accuracy, but the tanh activation used by LeNet-5 had the highest and therefore was used for the model. Classification on the total data set (excluding those removed by the filtering step) labeled 904,550 images as elliptical galaxies and 757,640 images as spiral galaxies.

1.3.2 Secondary Classification

Following the classification described in Section 1.3.1, the set of images predicted as spiral was shown to contain a significant number of "ghosts", or unclean images. The CNN classifier interpreted the unclean images as patterns of spiral features, leaving the elliptical predictions relatively clean.

To remove these ghosts, we constructed a second deep CNN to separate them from the true spirals. The architecture of this model is simpler than the first, using three convolutional layers with filter sizes of 7×7×8, 5×5×32, and 3×3×64, ReLU activations, and a single SoftMax output layer. Between the convolutional layers are max pooling layers that each reduce the input dimensions by half. The model uses the Adam optimizer and categorical cross entropy for loss.

For training, several hundred ghost images were initially selected from the set of galaxy images that were mistakenly predicted as spirals, and an equal number of spiral galaxy images were randomly selected from the original spiral training set. These images were divided into 70% training and 30% testing subsets as before. The model converged during training, and the images originally labeled as spirals were further classified into true spirals and ghosts. This process was repeated several times by selecting additional training images from those labeled as "ghosts" until the size of the training set reached 4,000 images. The final iteration of this classifier identified a total of 63,854 images as "ghosts" ($\sim$ 7.8%), removing them from the set of spiral galaxies.

1.4 Results

The application of the methods described in Section 1.3 to the Pan-STARRS images described in Section 1.2 provided a catalog of 1,662,190 galaxies. The catalog is accessible through a simple CSV file that can be downloaded at https://figshare.com/articles/PanSTARRS_DR1_Broad_Morphology_Catalog/12081144. Each row in the catalog is a galaxy and includes the Pan-STARRS object ID of the galaxy, its right ascension, declination, and the probability of the galaxy to be spiral or elliptical as estimated by the SoftMax layer of the CNN as described in Section 1.3. Figure 1.2 shows the number of galaxies available after applying a threshold to the output of the SoftMax layer of the model.

The catalog includes 904,550 galaxies identified as elliptical and 757,640 identified as

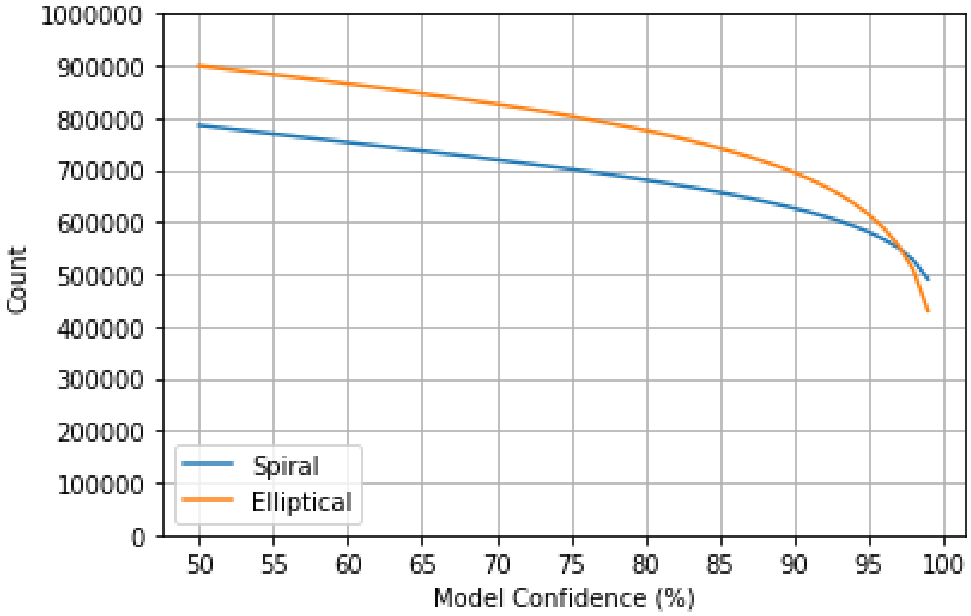


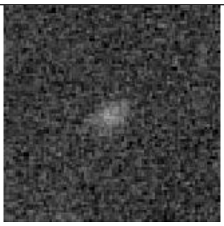
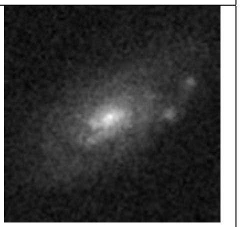
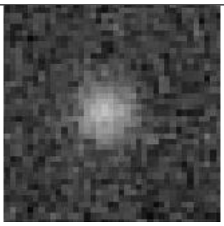
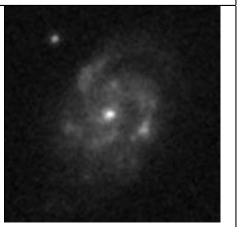
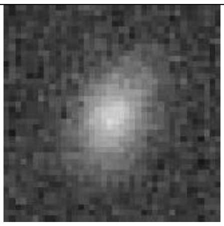
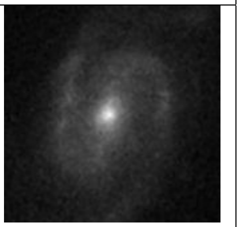
Figure 1.2: Number of spiral and elliptical galaxies remaining when keeping only those at or above a certain model confidence.

spiral. It should be noted that the annotation of a galaxy as an elliptical galaxy means that no spiral features were identified. However, the ability of an algorithm or a person to identify spiral features largely depends on the ability of the optics to provide a detailed image. Therefore, the identification of a galaxy as elliptical does not necessarily guarantee that the galaxy does not have spiral features, but that the optics cannot identify such features[43]. For instance, Table 1.2 shows examples of galaxies images by Pan-STARRS and the same galaxies imaged by Hubble Space Telescope (HST). These galaxies do not have visible spiral arms in Pan-STARRS, while the arms are clearly seen in the HST images.

1.4.1 Comparison to existing SDSS catalog

In the absence of a large manually annotated galaxy morphology catalog of Pan-STARRS galaxies, the evaluation of the consistency of the annotations was done using annotations of SDSS galaxies that were also imaged by Pan-STARRS. The largest catalog of broad morphology of SDSS galaxies is Kuminski and Shamir (2016)[30], with annotation of $\sim 3 \cdot 10^6$ galaxies. We will henceforth refer to this catalog as KS16. Although SDSS is a different sky

Table 1.2: Galaxies imaged by both Pan-STARRS and HST. While the Pan-STARRS images do not show clear spiral arms of the galaxies, HST shows that these galaxies are clearly spiral, and the arms can be identified.

Coordinates	Pan-STARRS	HST
$(150.165^o, 1.588^o)$		
$(150.329^o, 1.603^o)$		
$(149.951^o, 1.966^o)$		

survey, the footprint of SDSS overlaps with the footprint of Pan-STARRS. Since the KS16 catalog is large, it is expected that some galaxies in it will also be included in the catalog of Pan-STARRS galaxies described in this paper.

To evaluate the catalog, the annotations were compared to the annotations of SDSS galaxies in KS16 with a high degree of model confidence. Since the images of KS16 are collected and processed by the SDSS pipeline, their object identifiers naturally do not match the identifiers of Pan-STARRS objects. Therefore, the objects were matched by their coordinates, with tolerance of 0.0001^o to account for subtle differences in measurements between the two telescopes. This produced 13,186 total matches with 1,961 having 90% or higher confidence in the KS16 catalog. Figure 1.3 shows the degree of agreement between the annotations of the galaxies in our catalog and the annotations of the galaxies in KS16 with high confidence level.

When comparing the accuracy of our catalog to the accuracy of KS16, their model was

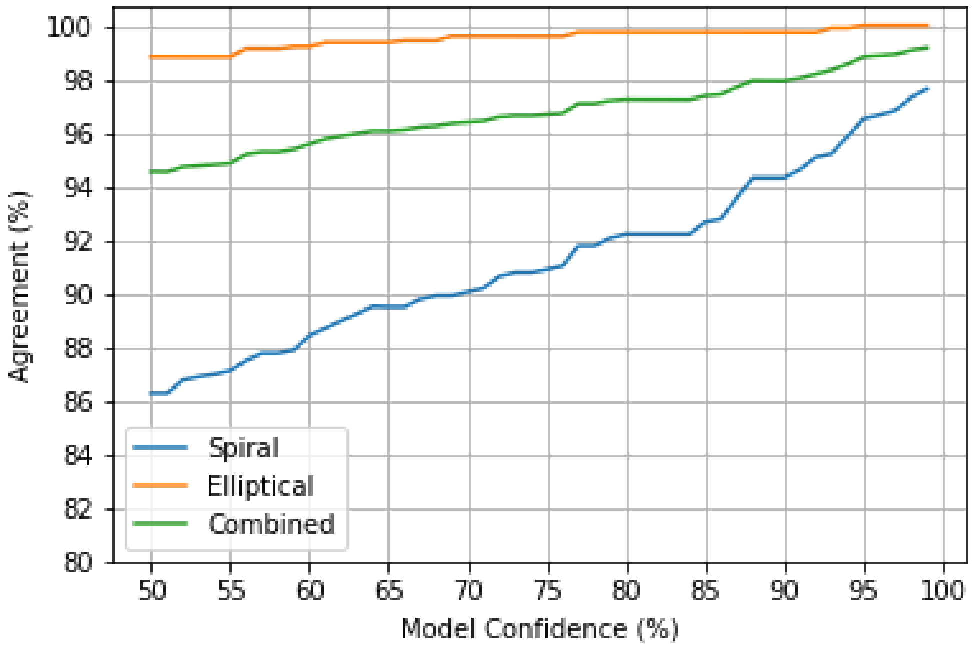


Figure 1.3: The proportion of predicted labels that, when restricted to a minimum confidence threshold, agree with the annotations in KS16. For example, restricting our catalog to labels with 90% confidence or higher results in approximately 98% agreement between the catalogs.

more accurate in identifying spiral galaxies, while the model used in our catalog was more accurate in the identification of elliptical galaxies. The algorithm used in KS16 is a "shallow learning" algorithm[44], which is a different paradigm of machine learning compared to the deep convolutional neural network used here. Shallow learning features such as textures and fractals might better reflect spiral arms, and therefore increase the ability of the algorithm to detect spiral galaxies. Elliptical galaxies are more consistent in shape than spiral galaxies, which can increase the performance of deep convolutional neural networks as their accuracy depends on the consistency of the images.

1.5 Conclusions

While digital sky surveys are capable of collecting and generating extremely large databases, one of the obstacles in fully utilizing these data is the difficulty of automatic analysis. Image data, and in particular images of extended objects, are more challenging to analyze due

Table 1.3: Examples of images that were misclassified by the model.

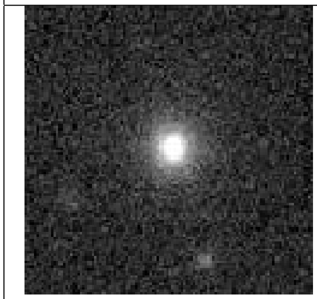
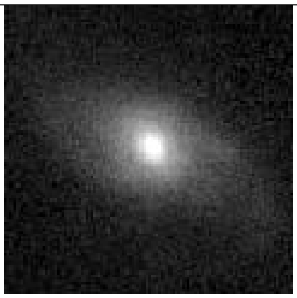
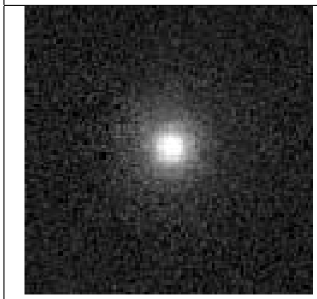
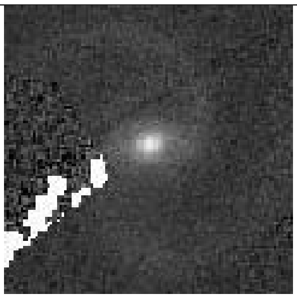
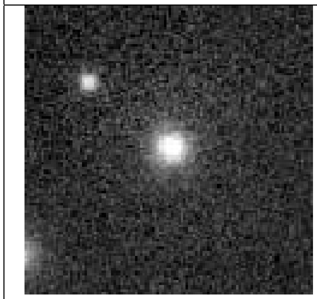
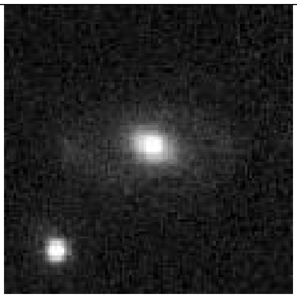
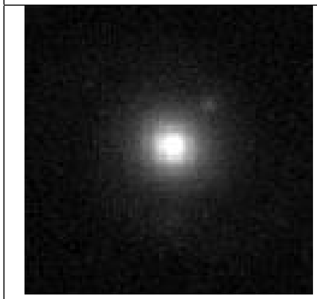
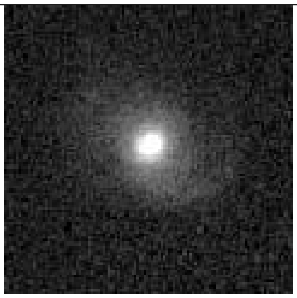
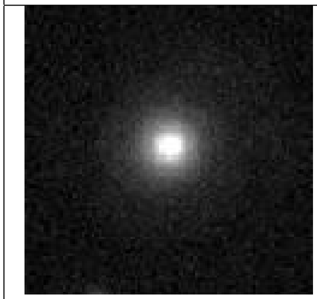
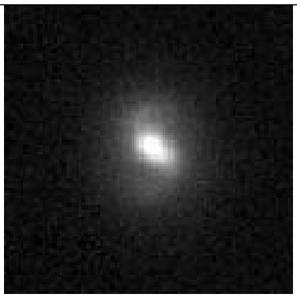
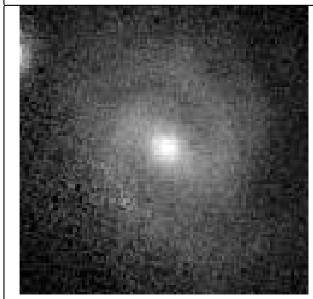
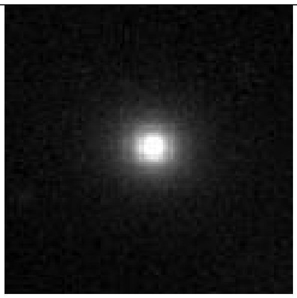
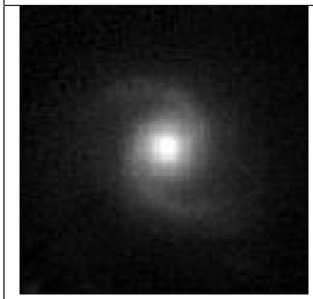
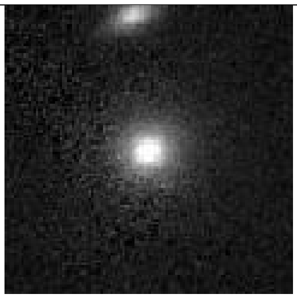
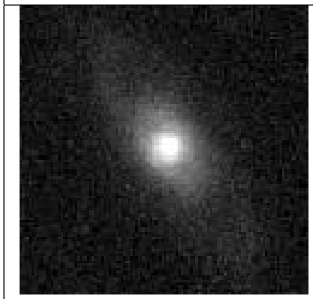
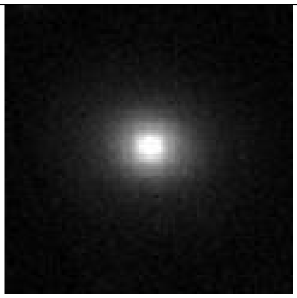
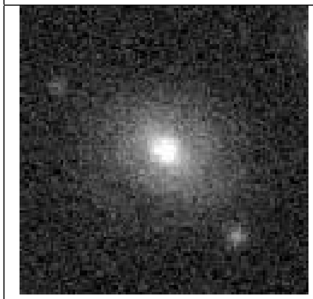
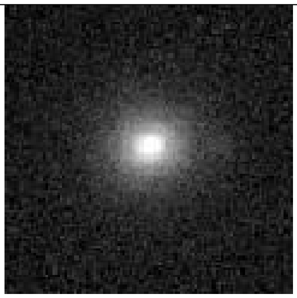
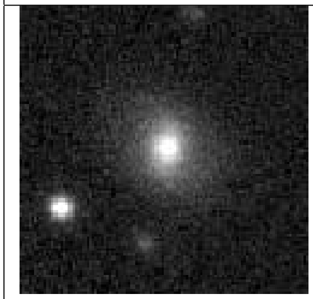
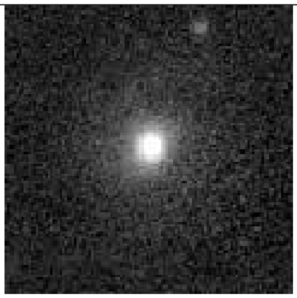
Misclassified as Spiral	Confidence	Misclassified as Elliptical	Confidence
	0.518151		0.601767
	0.561493		0.609336
	0.594011		0.645543
	0.767787		0.749332
	0.911469		0.764789

Table 1.4: Examples of images that were classified correctly by the model.

Classified as Spiral	Confidence	Classified as Elliptical	Confidence
	0.999981		0.999997
	0.998815		0.879958
	0.971841		0.756897
	0.744608		0.678047
	0.516342		0.563774

to the complex nature of the object's size and shape. Here we created a catalog of Pan-STARRS galaxies classified by their broad morphology into elliptical and spiral galaxies. The confidence in the predicted label provided by the SoftMax layer allows researchers to select a subset of the catalog with higher quality annotations. The catalog is available in the form of a CSV file at https://figshare.com/articles/PanSTARRS_DR1_Broad_Morphology_Catalog/12081144.

As space-based missions such as Euclid and ground-based missions such as the Rubin Observatory are expected to generate high volumes of astronomical image data, computational methods that can label and organize real-world astronomical images are expected to become increasingly pivotal in astronomy research. Such methods can provide usable data products at a rate far beyond traditional methods. While convolutional neural networks have demonstrated their ability to classify galaxies by their morphology, a practical solution needs to handle noise, bad data, and inconsistencies that are typical to large real-world datasets. As shown in this paper, the deep neural network alone may not be sufficient to provide useful data products. Instead, a combination of several algorithms that complete a full data analysis pipeline was needed. With the increasing robustness of such systems, it is also expected that protocols that combine multiple neural networks and filtering algorithms will be used to provide detailed morphological information.

The processing was done by first downloading the galaxy images to another server, and the analysis of the data was done on that server. The reason for using that practice is because the data analysis is based on solutions designed specifically for the task of galaxy annotation, and not on "standard" tasks provided by common services such as CasJobs[45]. Although the smaller JPG images were used, downloading all images still required a substantial amount of time. Using the more informative FITS images would have increased the required time to download the data by an order of magnitude, and analyzing data of much larger digital sky surveys such as the Rubin observatory will become impractical using this practice. Therefore, future surveys might provide users not merely with certain specific pre-designed tasks, but might also allow processing time for user-designed programs to access the raw data without the need to download it to a third-party server.

Chapter 2

SVMnet: Non-parametric Image Classification Based on Convolutional Ensembles of Support Vector Machines for Small Training Sets

This chapter was published in Goddard, H., Shamir, L., SVMnet: Non-parametric image classification based on convolutional ensembles of support vector machines for small training sets, *IEEE Access*, 10, 24029-24038. IEEE, 2022

2.1 Introduction

Deep convolutional neural networks (DCNNs) are powerful tools for multiple tasks of automatic image analysis, demonstrating paramount success and consequently gaining substantial popularity over the past decade. By analyzing the pixels directly, CNNs can be applied to various types of image content without the need to develop task-specific algorithms, and can easily be applied to a broad range of domains with excellent performance[46].

One of the major weaknesses of modern DCNNs is their dependence on a large set of examples for training. Cutting-edge DCNNs can have hundreds of layers, each with thousands of trainable parameters. For instance, the common ResNet-50[47] contains over $2{\cdot}10^6$ artificial neurons. Therefore, to achieve meaningful performance and avoid overfitting, DCNNs normally rely on relatively large training sets.

Training DCNNs normally requires large datasets of labeled ground truth images. Commonly used datasets include benchmarks such as ImageNet or the Modified National Institute of Standards and Technology (MNIST) dataset of handwritten characters. These benchmark datasets provide tens of thousands of images with high-quality annotations for training deep CNNs, and are commonly used for testing their performance. However, in many cases of real-world image classification problems, large datasets of clean, labeled ground truth are not available.

For instance, in the biomedical domain machine learning is often used for the purpose of image-based diagnostics[48]. However, the acquisition and annotation of each image can require the use of costly medical instrumentation, technicians, and medical staff who can annotate each sample manually[49;50]. Acquiring a single MRI image can take 30 minutes or more of using the instrument, excluding the time required to prepare the subject. The cost involved in the acquisition of such image is non-negligible. Even when using a quicker and less expensive imaging such as x-rays, the annotation of the data normally requires two or more trained experts, and the time they invest in the annotation is both expensive and time-demanding. That bottleneck has substantial impact on the ability of researchers in the medical domain to acquire large datasets.

Additionally, in the biomedical domain, human protection procedures and protocols are required for the acquisition of each sample, making the preparation of large datasets less practical. Therefore, biomedical image datasets are normally far smaller than the modern datasets commonly used to train DCNNs such as MNIST or ImageNet. In some cases the acquisition of images can involve substantial pre-processing, preparation of slides, staining, and imaging of each slide[51]. That is often the case when performing histological analysis for the purpose of diagnostics using machine learning[52–54].

Rare cases can also make it difficult to acquire a suitable training set[55]. For instance, to prepare an image-based diagnostics system that can automatically detect a rare clinical condition, a sufficiently large number of images of that rare case is required. In many cases, even when the resources are not limited by neither time nor cost, a sufficiently large number of cases is difficult to find.

Clearly, situations in which the dataset is small are not limited to the biomedical domain. Scientific experiments that require annotated data are very often limited by the resources required to annotate them. One of the solutions that the scientific community proposed is the use of crowdsourcing[56–58]. By crowdsourcing, non-expert volunteers can help annotating images or other data. With a large number of volunteers, the annotation of large datasets becomes feasible, and the resulting annotated datasets can be used to train machine learning systems. However, such crowdsourcing campaigns can take several years to complete[59], and are subjected to human error and human perceptional bias[60]. In many cases the annotation requires an expert, and the task cannot therefore be performed by anonymous untrained volunteers. In practice, experimentalists are often limited in their ability to utilize crowdsourcing for annotating a specific dataset.

The need for a large number of training samples is a practical downside of DCNNs, making them difficult to use optimally in many real-world cases. A common solution to increasing the size of the training set is data augmentation, in which different modifications of the images in the original dataset can create more training samples. However, that strategy can also lead to biases by overusing the same examples. In some cases transfer learning can be used to fine-tune neural networks using pre-trained models. Transfer learning is a proven tool to reduce the required training set size, but for domains with very small datasets for fine-tuning, the pre-trained models may remain too sensitive to their original task.

The problem of small training sets has been addressed in the past by using previous knowledge for few-shot training[55] and even one-shot training[61–65]. These methods reduce the number of required samples dramatically to as low as just one, but also require prior knowledge that is not necessarily available in all cases. Other related solutions include 3-D octave convolution with the spatial-spectral attention network[66] or deep attention graphs[67] for the problem of hyperspectral image classification.

This paper explores a new form of non-parametric image classification in cases when the number of samples is limited. Based on an ensemble composition of support vector machines (SVMs), the method can work with no prior knowledge, in a similar manner to "standard" supervised machine learning. Inspired by CNN architecture, SVMnet utilizes a large number of small SVMs to quickly analyze image patches, structured in layers that allow for stacking or custom ensemble techniques. An SVM[68] is less sensitive to high-dimensionality feature spaces[69–71], and can learn from a relatively small number of training samples[72–75] compared to other supervised machine learning approaches.

The primary advantage of the proposed method is that it outperforms the common DCNN architectures in cases when the number of labeled training images is small. As discussed above, such cases are not uncommon in real-world settings. Another advantage of the method is its much shorter training time compared to the time required to train deep neural networks.

2.2 Architecture of SVMnet

The proposed SVMnet architecture is designed as a stacked ensemble of numerous simple SVM classifiers organized into one or more layers. Each layer is an array of SVMs which functions similarly to a convolutional layer in a CNN. Each SVM in a layer is independent and all are assigned an equal-sized patch of the layer's input, referred to as a window. Variable stride length and padding, as described in Chapter 2 of[76], are specified as hyperparameters. Each input to the following layer is the output of one SVM.

When a layer is evaluated, each SVM in the layer is trained on ground truth labels. The input to the SVM is the flattened portion of each input image that is within the SVM's window. Each pixel channel within the window is essentially treated as one input feature. For instance, a 5×5 window would create a 25-feature SVM for grayscale input and a 75-feature SVM for 3-channel RGB input. During this step, the SVMs may be given weights based on the accuracy of the fit, used for ensemble classification. Each SVM then predicts a class label or a vector of class probabilities for its window of each input, creating an input tensor for the next layer.

Fig. 2.1 shows a simple layer in SVMnet. Each node in the layer is one SVM, trained using the ground truth labels for the input samples. The weights are determined based on the classification accuracy of the SVM compared to the ground truth of the training set. The weight function is configurable and will be described later in this Section.

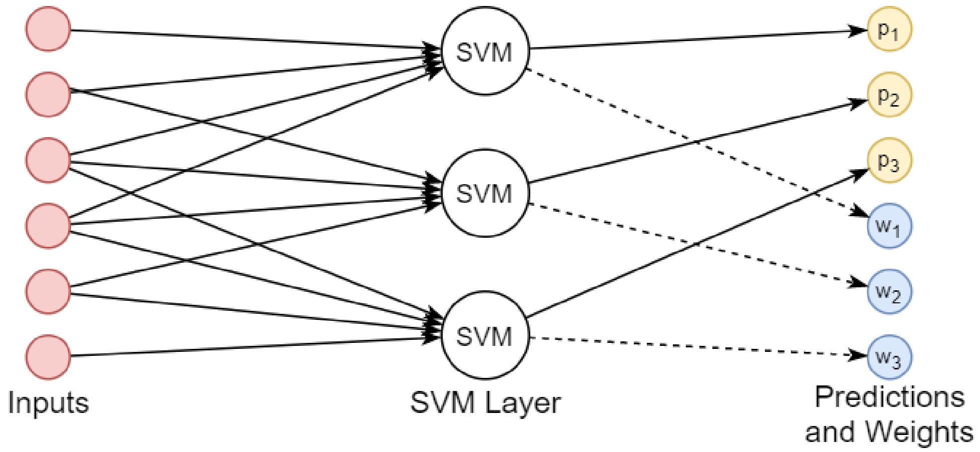


Figure 2.1: Example of a simple weighted layer of SVMnet. Each node in the layer is an SVM, trained with a subset of the inputs (pixels). Weight outputs are optional for a given layer.

To produce one class label for each input, SVMnet may perform a weighted vote after the final layer. This vote combines the results of the final layer by treating each value as a vote for that class label. If the final layer is weighted, these are used to weigh the votes in favor of SVMs with higher accuracy.

$$S_c = \Sigma_i \eta(A_i)[P_i = c] \tag{2.1}$$

The total voting score S_c of each class c is calculated by (2.1), where A_i is the accuracy score of SVM i in the final layer, η is the weight function, and P_i is the class label predicted by SVM i. That is, if the predicted label P_i of SVM i is class c, the weighted score $\eta(A_i)$ is added to the vote for that class. The weight function emphasizes the predictions of the SVMs with higher accuracy during training. The class that has the highest score S_c is chosen as the predicted label by the model for the given sample. The weight function η is configurable and in our experiments is defined as $\eta(x) = x^2$, where x is the classification accuracy of the SVM determined during training.

While the layers support arbitrary estimators, here we use only support vector machines (SVM), hence the name SVMnet. The SVMs are trained with a Radial Basis Function (RBF) kernel[77] and scaling gamma value, and they continue to iterate until convergence with a 0.001 tolerance. The ability to choose different estimators in each layer can be compared to the ability to use different activation functions in the layers of neural networks.

Fig. 2.2 illustrates one possible two-layer SVMnet architecture. Each SVM in the first layer analyzes a specific patch of each image and is fitted independently against ground truth labels. These SVMs then produce a vector of class probabilities for the same pixel region which forms the input matrix for the following layer. The SVMs in the second layer are fitted on a region of these probabilities and predict a class label for the image. These labels are then tallied in a final vote to produce one label for the input. The motivation for multiple layers is that layers after the first can in essence learn which of the SVMs in their window are more accurate or "trustworthy", as their predictions are being compared to ground truth labels in each layer.

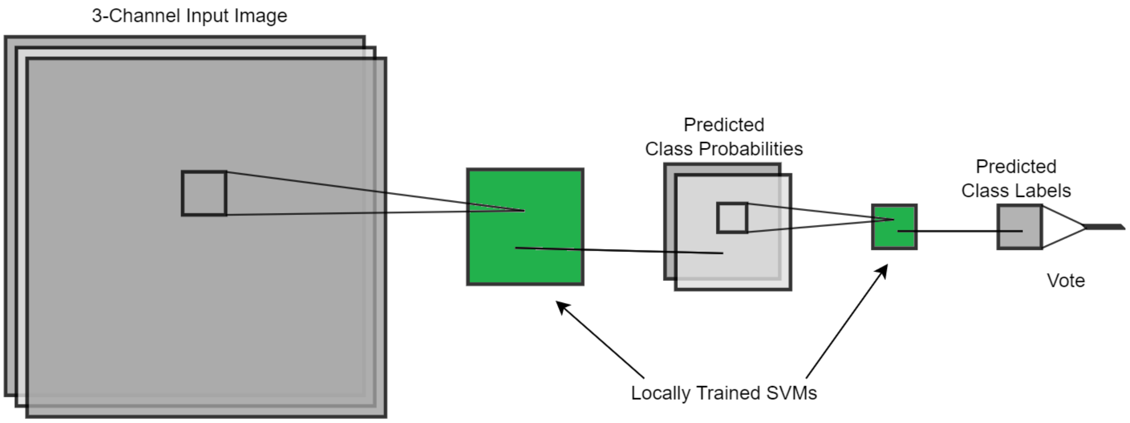


Figure 2.2: Example SVMnet architecture containing two SVM layers (in green) and a class label vote. Each SVM is trained on a patch of the layer's input. An $n \times m$ SVM layer produces $n \times m \times d$ output ($d \geq 1$).

2.2.1 Dropout

Not every patch is expected to produce a well-informed SVM. Some regions of the images, particularly towards the edge, often lack the details necessary to distinguish samples from each other. This can cause the outputs of these SVMs to act as noise in a vote tally. Even with the expected low accuracy score of the SVM depressing the weight of its vote, if the low-information regions are large then enough inaccurate votes may overwhelm the more accurate votes. To help prevent this, a dropout system is implemented for the vote tally.

When using dropout, which SVMs to drop are calculated when fitting SVMnet. First, the SVMs are ordered from the highest weight to the lowest. Votes are then cumulatively tallied one SVM at a time with the accuracy of the votes measured between each tally. SVMnet then finds the global maximum accuracy of the cumulative tally. This marks the point where including the votes of the less-accurate SVMs lowers the overall accuracy of the tally, so those SVMs are marked for dropout and are not included in the final vote. When the model is used to make predictions, the vote will only include the outputs of the SVMs that contributed to the most accurate tally.

In most cases during testing, automatic dropout resulted in equal or better performance than without dropout, as the least informative regions of the image were ignored. However, as with all hyperparameters, performance sometimes decreased and required fine-tuning. In each of the experiments described in Section 2.3, the SVMnet model presented is the one with the highest-performing hyperparameters among the combinations tested.

2.2.2 Formal definition of SVMnet

SVMnet can be defined formally as a 4-tuple as shown by Equation 2.2:

$$SVMnet = (T, C, S_0, \Phi), \tag{2.2}$$

where T is the topology of the network, C is the initial constants, S_o is the initial state of the network, and Φ is the set of SVM classifiers. The components that make the SVMnet are defined by Equation 2.3.

$$\begin{aligned} & T = (V, E) \\ & C = \{W, \Theta\} \\ & S_0 = \{\psi_i\} \\ & \Phi = \{(\Xi_i, \gamma_i, C_i)\} \end{aligned} \tag{2.3}$$

The topology $T = (V, E)$ reflects the structure of the network, where V is the nodes and E is a set of connections $E_{i,j}$ between the nodes $V_i \to V_j$, where V_i and V_j are two connected nodes. A pair of nodes $V_i, V_j \in V$ can have one or zero connections between them. Like in artificial neural networks, the topology $T = (V, E)$ determines the number of layers, number of nodes per layer, and the kernel size.

The constants C include the thresholds W, which are the threshold values used for ignoring the output of an SVM classifier as explained in Section 2.2.1. Each connection $E_{i,j}$ between two nodes is assigned with a threshold $W_{i,j}$, which determines whether the output of the SVM node i is used as an input to SVM node j. Unlike neural networks, in SVMnet these threshold values are constants, as they are not changed during training. Whether these threshold values impact the analysis depends on the consistency of the input, such that an inconsistent SVM node is ignored if its consistency observed using the ground truth training data does not meet the threshold. The use of these thresholds is explained in detail in Section 2.2.1. Another constant is Θ, which is the number of classes.

The initial status of the network S_0 is a collection of SVM hyperplanes ψ, such that the hyperplane ψ_i is the initial hyperplane of the SVM in node i. The hyperplanes are changed during the training of the SVMnet, as the SVMs learn from the data.

The set of SVM classifiers Φ is defined by $\{(\Xi_i, \gamma_i, C_i)\}$, such that each SVM classifier Φ_i is defined by its kernel Ξ_i, its gamma parameter γ_i, and its C parameter C_i. In the implementation shown in this paper all SVMs are defined by the same parameters, but other implementations are also possible in which different SVMs have different kernels or other parameters.

2.3 Experimental Results

To test the efficacy of SVMnet compared to a "conventional" CNN, several experiments were performed using common, relatively small datasets. The purpose of SVMnet is not to outperform CNNs in the general case, but to achieve higher accuracy when the number of labeled training images is limited. Therefore, the experiments were made with different sizes of training sets to compare the classification accuracy as the training set increases.

The performance of the SVMnet was compared to the performance of residual network, or ResNet, models with 18, 34, and 50 layers[47]. ResNet is a powerful architecture that was designed to reduce the number of required training samples for deep learning tasks and has demonstrated excellent efficacy in image classification. Each ResNet model was compared when trained from scratch and when fine-tuned using pretrained ImageNet weights. Following the practice in[47], the final convolutional layer is followed by a global average pooling layer, then by a single fully-connected layer with softmax activation and as many units as class labels in the respective task. Models were trained using stochastic gradient descent (SGD) optimization with a linearly decaying learning rate (given by $0.999(1-s/2)+0.001$ where s is the training step) and Nesterov momentum of 0.9. The models were trained for a maximum of 200 epochs but were stopped early if the loss on the validation dataset did not improve by at least 0.01 over 20 epochs. The number of epochs is limited in order to keep the ResNet training times comparable to SVMnet. The resulting accuracy and training time for each model was averaged over 5 repetitions of each experiment.

While the height and width of inputs can be adjusted for ResNet, the architecture always expects 3-channel RGB color images. Grayscale images were modified for use by ResNet by duplicating the pixel values into three equal channels. This approach was used in Section 2.3.3 and Section 2.3.4. Before training and classification by ResNet, images were also passed through a preprocessing filter provided by the Keras library to prepare the data for ResNet models. All inputs were normalized by dividing by the mean and subtracting the variance before being used to train SVMnet. For RGB color inputs, the images were normalized per-channel.

All experiments and analysis presented in this section used the same hardware environment. SVMnet was parallelized across 16 cores of Intel Xeon Gold 6130 CPUs, and ResNet models were trained on an nVidia GeForce GTX 2080 GPU.

2.3.1 COIL-100 Object Recognition

Columbia Object Image Library (COIL-100) is a common dataset used for basic object recognition[78]. It contains RGB color images of 100 different objects, each photographed 72 times at 5° increments about the vertical axis. Background details were removed in all images and the objects are centered and enlarged to fill the frame. Some objects contained in this dataset include coffee mugs, small toy cars, and various fruits and vegetables.

The SVMnet in this experiment used one layer with a 25×25 window (giving each SVM 1875 input features) and a stride length of 7, followed by a weighted vote with dropout. The SVMnet and ResNet models were fitted with 100-500 training images in increments of 100, each controlled to have an equal number of samples for each object. A separate subset of 200 images was used as validation data for ResNet models.

Fig. 2.3 shows the results of this experiment. When fitted on the smallest training set, containing only one example per object, SVMnet correctly predicted labels for over 60% of the remaining images. With the same training set, ResNet-50 showed about the same accuracy and only pretrained ResNet-34 exceeded SVMnet; however, SVMnet was significantly faster to train in all cases.

2.3.2 Imagenette

Imagenette is a fairly small, 10-class subset of the ImageNet dataset[79]. Several versions of this dataset exist; here we use version 2 of the 160 px dataset with noiseless labels. Many of these images are rectangular with their shortest side scaled to 160 px. In this experiment, we symmetrically zero-pad each image along its shorter axis to make it square, then downscale the images to have the same dimensions of 160×160 px.

The SVMnet used here contains one layer with a window size of 22 and stride length

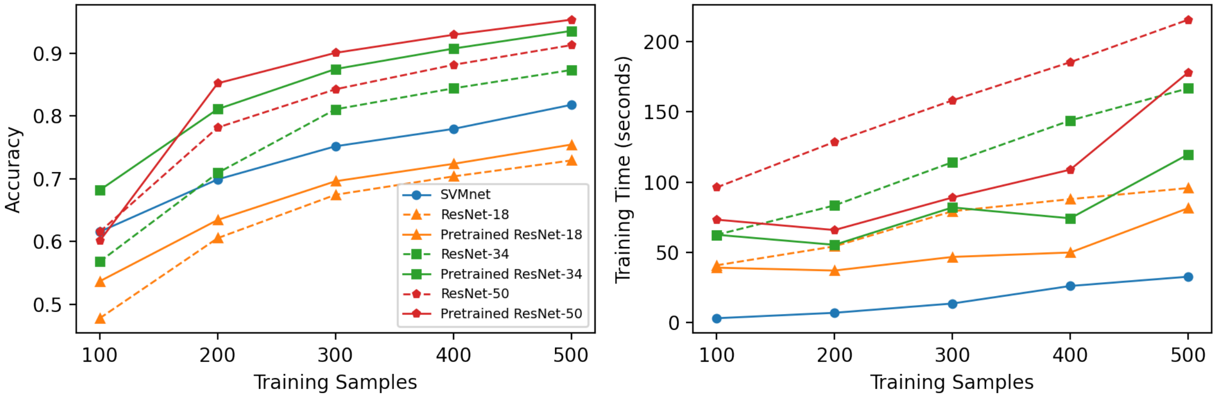


Figure 2.3: Test-set accuracy (left) and training time (right) of SVMnet and ResNet on COIL-100 images when fitted with different training set sizes.

7, followed by a weighted vote with no dropout. Imagenette is pre-divided into training and testing subsets containing 9,469 and 3,925 images, respectively. Models were trained using 20, 40, 80, 160, and 320 images from the provided training set and evaluated using the provided testing set. An additional 100 images were selected from the training set as validation data for the ResNet models.

Fig. 2.4 shows the results of this experiment. SVMnet achieved higher accuracy than all ResNet models for all training sets except the largest, where the ResNet-50 model pretrained with ImageNet weights improved drastically. The generally low accuracy of these models could be explained by the method used to conform each image to the same dimensions, which introduces a significant amount of empty space in many images. However, even under these conditions, SVMnet attained the highest accuracy in the least time for the smaller training sets.

2.3.3 COVID-19 Radiography

During the COVID-19 pandemic, machine learning techniques have been applied to various kinds of data to assist the medical community in making accurate diagnoses[80–82]. During the early stages of a disease outbreak, diagnostic data is expected to be limited or sparse, making it difficult to train most kinds of machine learning models. A type of model capable

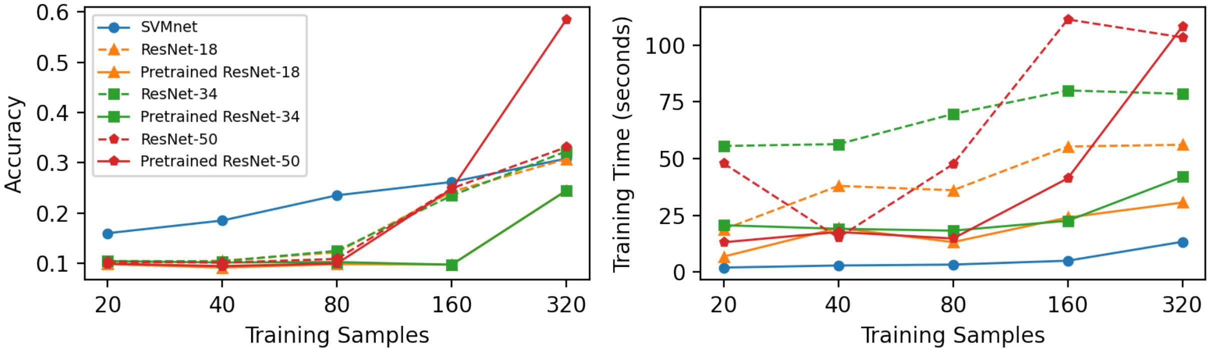


Figure 2.4: Test-set accuracy (left) and training time (right) of SVMnet and ResNet on Imagenette when fitted with different training set sizes.

of learning from a small number of samples would be the most effective in this time frame.

Here we apply SVMnet to a database of chest x-ray images from healthy patients and patients diagnosed with COVID-19[83;84]. In this experiment, only the images labeled as ”Normal” and ”COVID” are used. Images were downscaled to 128 × 128 pixels (approx. 43% of the original size). An equal number of images were selected from each class, totaling 7232 samples. Models were fitted with 10, 20, 50, 100, and 200 training samples, with 50 separate images used as validation data for the ResNet models. The SVMnet uses two layers: the first with window size 19, stride 7, and class probability outputs; the second with window size 5 and stride 5, followed by an unweighted vote. During the architecture experiments described in Section 2.3.7, the 2-layer SVMnet was shown to outperform the 1-layer models for this dataset.

Fig. 2.5 shows the results of this experiment. SVMnet was able to correctly label between 64% and 78% of unseen x-rays depending on the number of training samples, but most ResNet models failed to make significantly accurate predictions. Only the 18- and 34-layer ResNet models trained from scratch approached the accuracy of SVMnet. Additionally, SVMnet was several times faster to train.

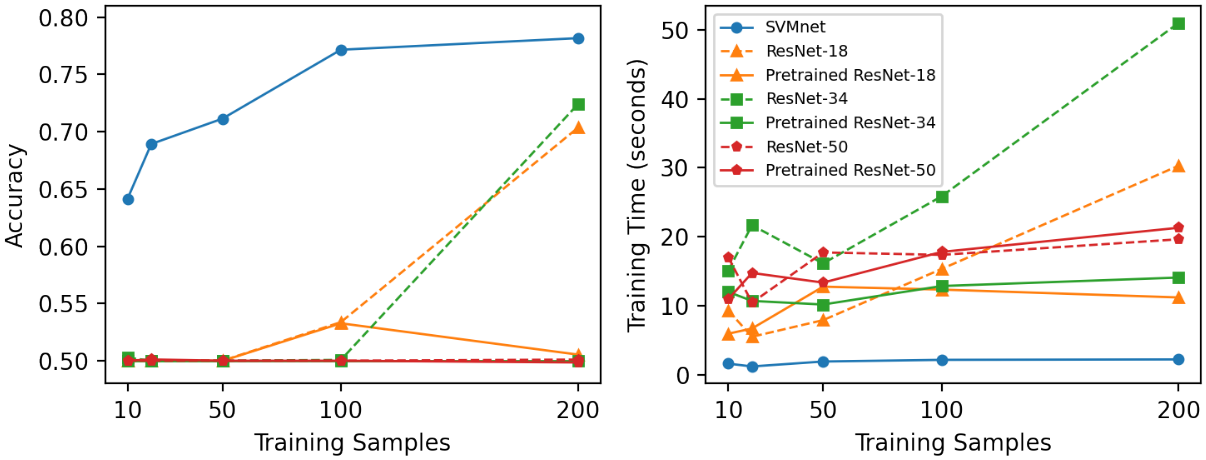


Figure 2.5: Test-set accuracy (left) and training time (left) of SVMnet and ResNet on COVID-19 chest x-ray images when fitted with different training set sizes. The accuracy of the ResNet models displays considerable overlap.

2.3.4 Astronomical image data

To test the performance of SVMnet on a current real-world image classification problem, a dataset of galaxy images from the Panoramic Survey Telescope and Rapid Response System (Pan-STARRS) was used. The dataset is made of galaxies separated into elliptical and spiral morphology. The galaxy images were taken from the catalog of Pan-STARRS galaxies classified by their morphological type[85].

An equal number of images were selected of each morphological type, totaling 26,732 samples. Each image is grayscale and has a dimension of 120×120 px. SVMnet and ResNet models were fitted with 10, 20, 40, 80, 160, and 320 training samples, with 200 separate images used as validation data for the ResNet models. The SVMnet uses one layer with a window size of 22 and stride 5, followed by a weighted vote with dropout.

Fig. 2.6 shows the results of this experiment. As the graph shows, SVMnet outperformed almost every ResNet model when trained with a relatively small dataset. The models generally improve as the training set grows, with several ResNets slightly overtaking SVMnet with the largest training set. In all cases, SVMnet finished training many times faster than all ResNet models.

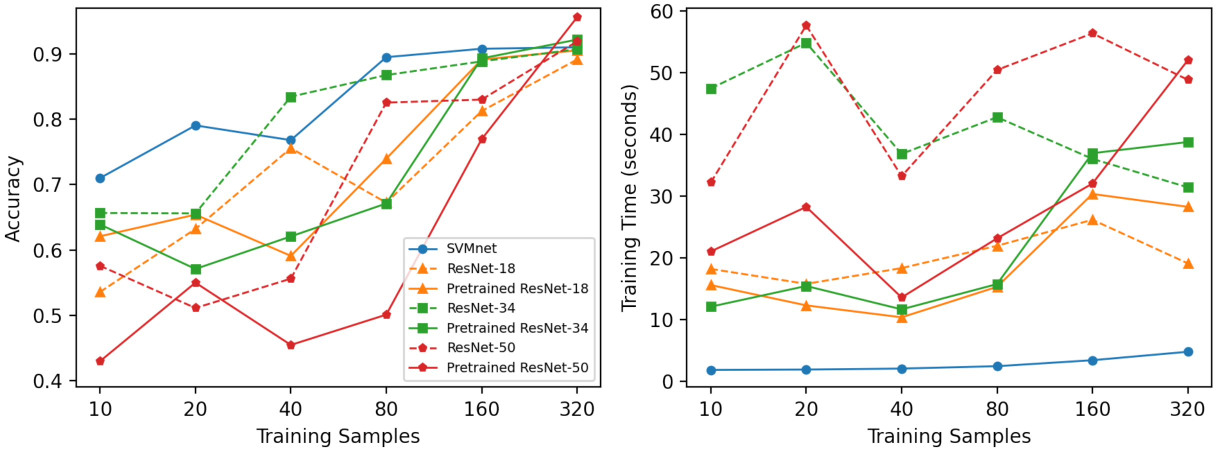


Figure 2.6: Test-set accuracy (left) and training time (right) of SVMnet and ResNet on Pan-STARRS galaxy images when fitted with different training set sizes.

2.3.5 WND-CHARM

To test a ”traditional” approach of using an SVM after extracting image features, we used the WND-CHARM open source feature set[44] combined with an SVM with linear kernel implemented through SVMLib. Table 2.1 compares the test set accuracy of WND-CHARM and SVMnet using the experimental datasets described earlier in this Section. WND-CHARM was trained on equal sized training subsets and consistently showed lower classification accuracy than SVMnet under the same conditions.

2.3.6 Computational complexity

The complexity of fitting an SVM is asymptotic and polynomial. For a training set containing n samples, the algorithm is dominated by either an n^2 term or an n^3 term based on the formulation of the problem[86]. Therefore, training a large number of SVMs can be a computationally demanding task, and can lead to substantial computational complexity during training.

The number of SVMs N in a layer receiving rectangular input with width I_x and height I_y is given by (2.4). The window size W (equivalent to the kernel size in other CNN literature), stride length S, and padding amount P in their respective dimensions follow from standard

Table 2.1: Comparison of the classification accuracy of WND-CHARM and SVMnet when trained on a small number of samples from four datasets.

COIL-100

	WND-CHARM	SVMnet
100	54%	62%
200	59%	70%
300	61%	75%
400	64%	78%

Imagenette

	WND-CHARM	SVMnet
20	11%	16%
40	13%	19%
80	16%	24%
160	18%	26%
320	21%	31%

COVID-19

	WND-CHARM	SVMnet
10	53%	64%
20	55%	69%
50	60%	71%
100	64%	77%
200	66%	78%

Pan-STARRS

	WND-CHARM	SVMnet
10	52%	71%
20	56%	79%
40	61%	77%
80	63%	90%
160	72%	91%
320	88%	91%

convolutional arithmetic. When using a square window on square input, the formula can be simplified to (2.5).

$$N = \left(\frac{I_x + 2P_x - W_x}{S_x} + 1\right) \cdot \left(\frac{I_y + 2P_y - W_y}{S_y} + 1\right) \tag{2.4}$$

$$N = \left(\frac{I + 2P - W}{S} + 1\right)^2 \tag{2.5}$$

Fitting a layer in SVMnet requires fitting N SVMs - a polynomial time operation. If the layer includes weights, then the SVMs must predict a class label for each input during the fit step, which scales linearly with the number of samples n. When using dropout as described in Section 2.2.1, SVMnet performs an additional step during training that scales linearly with n. Thus, fitting SVMnet is dominated by the polynomial fit time of the SVMs. This relationship can be observed experimentally in Fig. 2.8.

CNNs can theoretically be trained infinitely, but there is a definitive point at which the SVMs within SVMnet converge. This places a soft upper bound on the training time of

SVMnet based on the tolerance parameter of the SVMs. Additionally, a firm upper bound may be placed on the number of iterations of the SVM algorithm, allowing for a shortened training time at the expense of some accuracy.

SVMnet trains multiple SVMs simultaneously using process-based parallelism and shared memory, greatly increasing its speed on typical multicore computers with minimal overhead. While this allows SVMnet to run quite easily on relatively inexpensive systems, the potential performance gain from extra hardware is minimal compared to the extreme optimization of CNNs for GPU devices.

While the training of SVMnet is slower than CNNs when the size of the training set becomes relatively large, SVMnet is designed for situations in which the size of the training set is small. Therefore, the computational complexity of the training is not expected to introduce a major obstacle in many real-world cases where the size of the training set is limited, and the time required for training does not necessarily explode to an unmanageable response time in the situations where SVMnet is most effective.

Inference time of image classification

Predicting a single class label of an image using SVMnet typically requires a large number of individual SVMs to predict a label followed by a vote tally. Despite its affinity for parallelization, this process is expected to take longer than the highly optimized matrix operations of a CNN. Table 2.2 compares the inference time of SVMnet and ResNet on images in the COIL-100 dataset.

Table 2.2: Comparison of the response time (in seconds) of SVMnet and ResNet to predict class labels for 1, 10, 100, and 1000 samples of the COIL-100 dataset.

	1	10	100	1000
SVMnet	2.36	2.66	3.81	24.2
ResNet-18	0.054	0.056	0.082	0.296
ResNet-34	0.060	0.060	0.098	0.410
ResNet-50	0.061	0.064	0.106	0.515

The comparison shows that SVMnet is significantly slower than ResNet for classifying samples, but the speed of classification is still practical for many real-world systems. The parallelization of SVMnet greatly reduces the time needed to make predictions, but the overhead of shared memory operations is significant in the case of few samples.

2.3.7 Architecture Comparison

As with CNNs, SVMnet can be configured into a variety of architectures which are expected to differ in performance depending on the classification task. Due to the high number of possible models, determining which is the most effective for a single task is non-trivial. In this section we show how a variety of SVMnet configurations were tested on the COIL-100 dataset to inform the choice of model used in Section 2.3.1. Similar methods were used to select the models for other datasets. SVMnet models with multiple layers were tested in the same manner.

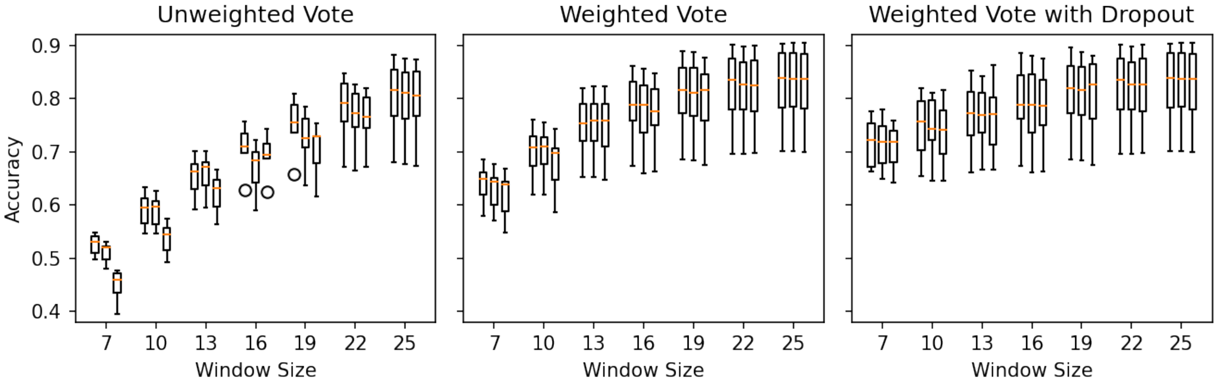


Figure 2.7: Prediction accuracy of one-layer SVMnet architectures fitted to COIL-100. Each group of three box plots represents the same window size with stride length 3, 5, and 7, respectively. Each box plot shows the distribution in model accuracy when using five training sets of 200-1000 examples.

Fig. 2.7 shows how the performance of a one-layer SVMnet changes with the window size, stride length, voting method, and number of training samples when fitted to COIL-100. Prediction accuracy improves in all cases as the window size increases but with diminishing returns. Increasing the stride length tends to lower accuracy when the window is small but incurs little to no penalty when the window is large. When the vote of an SVM is weighted, model accuracy improves in all cases compared to an unweighted vote; performance increases further when using dropout as described in Section 2.2.1. This effect is more significant when the window size is small.

Fig. 2.8 shows how the time required to fit SVMnet on COIL-100 changes with the number of SVMs (see Equation 2.5) and the number of features for each SVM (in this case equal to $3W^2$). Since increasing the stride length significantly reduces the number of SVMs in the model, an SVMnet with large windows can still be trained quickly with only a minor increase in stride without sacrificing accuracy. Each SVMnet in this experiment was trained in parallel using 16 CPUs.

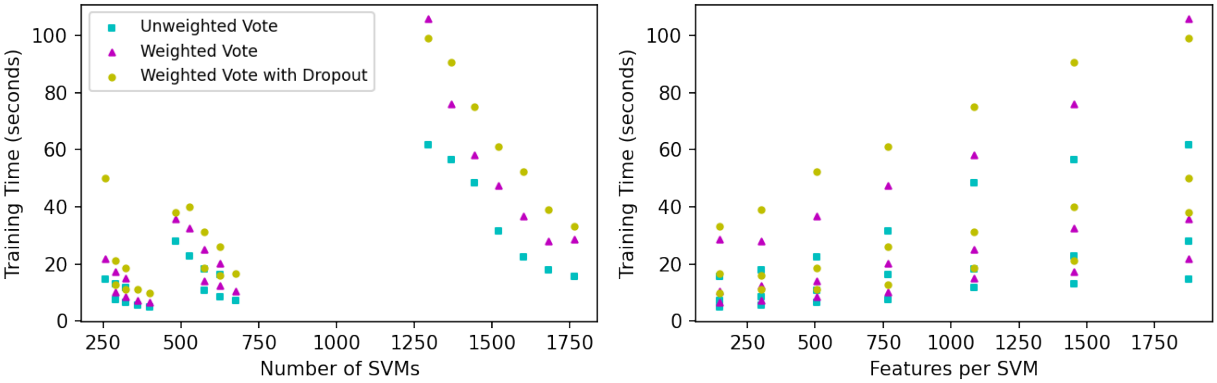


Figure 2.8: Training times for one-layer SVMnet architectures fitted to COIL-100.

2.4 Conclusions

Deep convolutional neural networks provide excellent performance in automatic classification of image data while eliminating the need to develop and tailor algorithms for specific image classification problems. With the availability of open source libraries, DCNNs have become the de facto first solution to image classification.

Here we explore one of the primary weaknesses of DCNNs, which is the need of a relatively high number of labeled "ground truth" samples for effective training of the network. While in computer science literature DCNNs are often tested on relatively large datasets such as MNIST or ImageNet, in many real-world problems a very large number of clean labeled samples that can be used for training is not available.

Medical datasets such as those prepared for the purpose of image-based diagnostics are difficult to prepare due to the long time required to assign a sample with a correct label[50], consequently leading to a high cost. Additionally, acquiring a radiograph can also require substantial resources, as medical image acquisition systems such as Magnetic Resonance Imaging (MRI) require expensive instrumentation and staff. Furthermore, the consent of the patient is required for the preparation of each sample. These limitations make large datasets of biomedical images substantially more expensive and difficult to prepare.

In many other cases labeled training samples are not available. For instance, when analyzing archaeological artifacts, the number of training samples is limited by the number of available artifacts, which is often a hard limit that cannot be easily changed. A typical size of such datasets is normally several hundred samples[87]. Using computer vision to analyze art[88] is limited by the number of paintings each artists created, which can be a firm limit, especially when the painter is no longer alive. These are obviously just a few examples out of many possible real-world situations in which the number of labeled samples is inherently small.

SVMnet aims at providing an effective solution for the numerous real-world situations in which the number of labeled image samples that can be used for training is limited. SVMnet utilizes the ability of an SVM to learn from a smaller number of samples compared to other machine learning approaches. The flexible structure of SVMnet allows it to learn directly from the pixel values, and to utilize different layers that correspond to the convolutional and fully connected layers in “conventional” deep neural networks.

Like DCNNs, SVMnet does not require the design of specific algorithms for a particular image classification problem. Therefore, SVMnet can be used for a variety of image data, as also demonstrated in Section 2.3. The proposed approach is structured as a network to take advantage of the stronger signal from neighboring pixels, similar to the core idea in the basis of CNNs. SVMs are known for their ability to learn quickly from relatively few training samples. By training many SVMs on small pixel regions across an image, this quick learning can be leveraged to extract much information from small sets of images in less time than it would take to fully train a deep neural network.

Complexity analysis shows that the training time for SVMnet scales more quickly with the number of input samples than DCNNs, suggesting that SVMnet might take substantial computational resources when trained using large datasets. However, SVMnet is designed for situations in which the labeled training set is relatively small. As shown in our experiments, the training time might not be a practical obstacle in many real-world situations in which SVMnet can be used. While computing is an available resource, and training SVMnet with a few hundred training samples scales with reasonable efficiency, clean annotations or rare training samples might in many cases be much more difficult to obtain.

The underlying structure used to create SVMnet is very flexible, providing several avenues for future work. For example, other kinds of machine learning algorithms can be used as sub-classifiers rather than solely SVMs. Constructing the layers with classifiers such as random forests or logistic regression may result in better performance for some datasets. These layers can be mixed in the same model as well, i.e. using one layer of SVMs followed by a layer of random forests, or using some combination of sub-classifiers in the same layer. Furthermore, there is much room for improvement in hyperparameter optimization. There is nothing constraining the sub-classifiers in a single layer to the same hyperparameter initialization, meaning that the optimization space for tuning the hyperparameters is much larger than in other kinds of machine learning models. Different methods for determining a class label from the output of a layer should also be considered, such as combining SVMnet with the proven efficacy of neural networks by using a multi-layer perceptron as a classification head.

SVMnet is not designed to become a general solution that can outperform deep convolutional neural networks such as ResNet-50. However, experimental results show that it is an effective solution for cases in which the number of labeled training samples is small. Since such cases are not rare, SVMnet can complement conventional machine learning methods by providing image classification in the cases where not many labeled training samples are available.

Chapter 3

Machine Learning and Neural Network Bias in Analysis of Photometric Data

This chapter was published in Goddard, H., Shamir, L., Neural network bias in analysis of galaxy photometry data, *18th IEEE International Conference on eScience.* IEEE, pp. 407-408, 2022. and Goddard, H., Shamir, L., Machine learning bias and the annotation of large databases of astronomical objects, *XXXII Astronomical Data Analysis Software and Systems (ADASS)*, 2022

3.1 Introduction

The information era has made a revolutionary impact on astronomy research. For instance, autonomous digital sky surveys have enabled the collection of very large astronomical databases, enabling unprecedented discovery power[1;89], and that trend is bound to continue[2;3;90].

Perhaps the primary data collected by autonomous digital sky surveys are the images of astronomical objects and their photometric information. For instance, the Panoramic

Survey Telescope and Rapid Response System[4;5] collected images and photometry data for over 1 billion astronomical objects. The Sloan Digital Sky Survey[91] has collected image and photometry information about an equivalent number of astronomical objects, and provided data that enabled more than 10^5 scientific papers[1]. While the Sloan Digital Sky Survey (SDSS) is a powerful sky survey, the Vera Rubin Observatory will collect an equivalent amount of data collected to date by SDSS once every three days. That is added to high-throughput space missions such as Euclid, also generating vast pipelines of data.

One of the primary outcomes of the data collected by digital sky surveys is the photometry data. Photometry data for each astronomical object includes its location, and measurements such as its color, brightness, shape, size, and more, based on the design of each specific photometric pipeline. Many photometric pipelines also add certain analysis of the data to provide useful information to the user. An example of such analysis is whether an astronomical object is a point source (e.g., a star) or an extended source (e.g., a galaxy) by applying a star/galaxy separation algorithm[92–95], normally to the photometry data. While that separation is not necessarily fully accurate in all cases, it provides useful information that can assist in obtaining better analysis without requiring the users of the data to apply the algorithms themselves.

Due to the very large amounts of data collected by modern autonomous digital sky surveys, manual analysis of these data becomes impractical, and automatic analysis is required. Given the complexity and high-dimensionality of the data, one of the common ways to approach the analysis of astronomical data is machine learning. By using existing supervised machine learning algorithms, researchers can annotate merely a small part of the data, and apply the algorithms to analyze large datasets by allowing the machine learning algorithm to extract complex rules driven by the data it was trained with.

One of the common uses of machine learning in astronomy is the photometric redshift, and numerous methods for photometric redshift have been proposed and used[96–99]. By using photometry information, the redshift of a certain extra-galactic object can be approximated to a certain degree of accuracy. That allows to determine the approximate distance of a

[1] https://www.sdss.org/science/

large number of objects without the need to analyze their spectra. While the photometric redshift is far faster than measuring the spectra, and can therefore scale to large astronomical databases, photometric redshift is not accurate, and it has been proposed that it can be biased[100–104].

Another common use of machine learning in the context of autonomous digital sky surveys is the automatic annotation of astronomical images. As autonomous telescopes can image millions or even billions of astronomical objects, machine learning algorithms can be used to annotate these objects automatically and determine their morphology or astronomical nature[23;105–114]. While these machine learning methods are becoming increasingly more common, deep neural networks have been shown to be subjected to biases[112;113;115–117], and certain bias has also been reported for astronomical images[118]. The purpose of this study is to test the existence of possible biases when using machine learning analysis of photometry data.

Data products generated using machine learning, and particularly deep learning, are common in astronomy[119]. These catalogs are often used by cosmologists, for example, to analyze the large-scale structure of the Universe. However, these researchers may be using these catalogs without being fully aware of the biases that can be present in the machine learning models used to produce them, which may be due to the perception that computers are more objective or trustworthy than human analysis. Machine learning models may carry forward biases present in their training data or learn improper patterns from noise in measurements[120]. The methods used to train a model may also introduce biases as a result of trade-offs between desirable properties and the target heuristic used to measure performance[121]. We hope to raise awareness of these issues and demonstrate the subtlety and unintuitive nature of machine learning bias.

3.2 Data

In this study, photometric data from two major digital sky surveys were used: the Sloan Digital Sky Survey (SDSS) and the Panoramic Survey Telescope and Rapid Response System

(Pan-STARRS). The data includes photometric information for objects that were identified as galaxies, and separated into spiral and elliptical classes. The morphological labels were taken from previous catalogs, including Kuminski and Shamir (2016)[122] for SDSS annotations and Goddard and Shamir (2020)[85] for Pan-STARRS annotations.

Both catalogs provide a level of certainty for the correctness of each label. To ensure that the data includes just clear annotations, SDSS galaxies were retrieved from data release 17 and limited to those whose morphology annotations had a confidence of at least 90%, while Pan-STARRS galaxies were retrieved from data release 1 with a minimum annotation confidence of 95%. These thresholds provide datasets with agreement of more than 98% compared to manual analysis[122].

The SDSS records photometric measurements for the u, g, r, i, and z bands, while Pan-STARRS uses the g, r, i, z, and y bands. Color features can be obtained by taking the difference of values in adjacent bands (subtracting the longer wavelength from the shorter one), such as $g - r$ and $r - i$. We compute these color features for the exponential, de Vaucouleurs, and Petrosian profiles in both datasets to use as inputs for our neural networks. In addition, we include the r-band magnitude and radius for each profile, as well as the radii containing 50% and 90% of the Petrosian flux. The resulting 20 features are summarized in Table 3.1.

Table 3.1: Feature vector summary for SDSS and Pan-STARRS datasets.

Profile	SDSS	Pan-STARRS
Exponential de Vaucouleurs Petrosian	$u - g$ color	
	$g - r$ color	$g - r$ color
	$r - i$ color	$r - i$ color
	$i - z$ color	$i - z$ color
		$z - y$ color
	r-band magnitude	r-band magnitude
	r-band radius	r-band radius
Petrosian	r-band radius containing 50% flux	
	r-band radius containing 90% flux	

Our SDSS data correspond to entries in the *Galaxy* view of DR17 with the clean photometry flag set and no missing/placeholder values (i.e. -9999) for fields used in our feature vector. After applying the 90% annotation confidence threshold, we are left with 247,427 galaxies, of which 126,110 are labeled as elliptical and 121,317 as spiral. Figure 3.1 shows the magnitude distribution of our SDSS dataset. Our Pan-STARRS dataset was collected in a similar manner from DR1 with the *primaryDetection* flag set. After filtering rows with missing values or annotation confidence below 95%, the dataset contains 991,518 galaxies with 549,292 labeled as elliptical and 442,226 as spiral. The magnitude distribution for our Pan-STARRS dataset is shown in Figure 3.2.

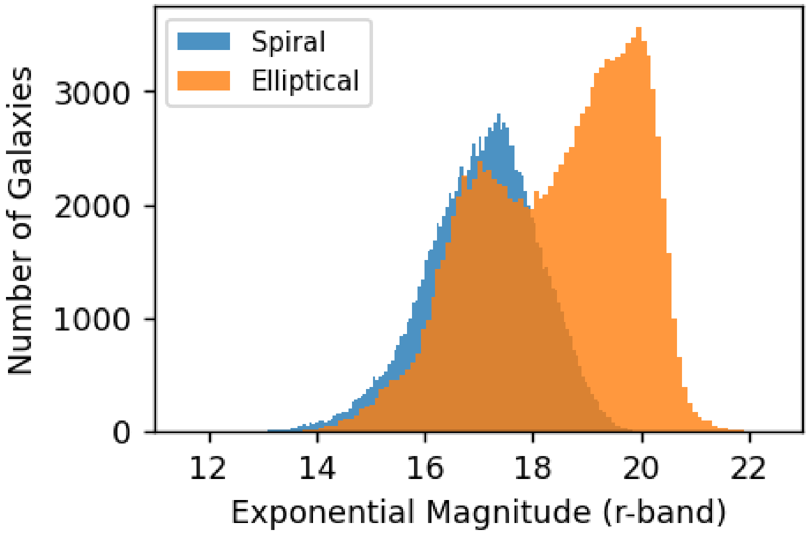


Figure 3.1: Histogram of SDSS galaxy magnitudes by morphological class.

3.3 Methodology

The application of neural networks to real-world problems relies on the ability of the trained network to properly generalize to unseen data. One of the ways this can fail is if the data used to train the model contains an underlying bias that distinguishes it in some way from other samples. The high complexity of neural networks enables them to recognize subtle features and patterns that are effectively invisible to human analysis, allowing biases to "sneak in" to even professionally curated training datasets[116;123].

In each experiment, we begin by programmatically determining a neural network architecture. Networks are assembled with one to four layers with each layer having either 16,

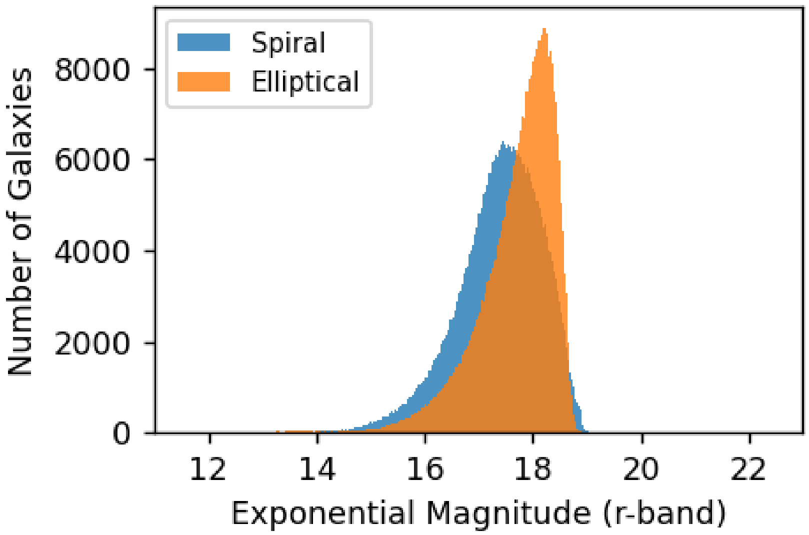


Figure 3.2: Histogram of Pan-STARRS galaxy magnitudes by morphological class.

32, 48, or 64 artificial neurons, creating 340 different architectures with varying width and depth. Each layer uses the rectified linear unit (ReLU) activation function and is strongly regularized with dropout[124] at a rate of 20% in the first layer and 50% in subsequent hidden layers. At the end of every model is a 2-neuron softmax output layer, corresponding to the two morphological classes in our datasets.

A random selection of $4 \cdot 10^4$ galaxies, divided into $3 \cdot 10^4$ for training and $1 \cdot 10^4$ for testing, is used to evaluate each candidate architecture. Spiral and elliptical morphological classes are represented equally in both the training set and the test set. The neural networks are trained for 100 epochs with 20% of the training set reserved for validation, a batch size of 100, and Adam optimization[125]. The architecture of the trained model that achieves the highest accuracy on the test set is used in all later stages of each experiment.

After selecting a network architecture, we analyze how the choice of training set affects the accuracy and distribution of predicted class labels. The dataset is divided into distinct sources from which training and testing data are sampled. Several neural networks are trained with different combinations of these sources (e.g. spiral galaxies from source A and elliptical galaxies from source B) and evaluated using a disjoint test set from each source. When a neural network is trained using data all from one source, we use that model's predictions for the test set from the same source as a baseline to compare the other models to. We refer to these predictions as *homogeneous annotations*, and they are assumed to be the least affected by biases caused by the choice of training set.

In Section 3.4, the predictions of the trained models are compared using the binomial test with the null hypothesis that the change in training set should not significantly alter the distribution of the model's annotations. That assumption is based on the fact that the data are taken from the same survey and that the annotations are taken from the same catalog. For each set of homogeneous annotations and morphological class, the expected probability of a correct annotation is the number of true labels for that class divided by the size of the test set. Then, for each model with a different training set, the number of true labels for that class predicted by that model is used as the observed number of successes for the binomial test. This method allows to take into account both the accuracy and class distribution of the generated annotations. We report the two-tailed p-value for each of these tests. A numerical example of this process is given in Section 3.4.1.

3.4 Generalizing to Different Areas of the Sky

Assuming that the Universe is homogeneous and isotropic, the distribution of spiral and elliptical galaxies as observed from Earth is expected to be the same regardless of the direction of observation. Here we test whether the part of the sky from which the training set is acquired can lead to unexpected differences in the distribution of elliptical and spiral galaxies in a catalog prepared by annotating the galaxies with an artificial neural network.

We select galaxies for this experiment from regions of the sky based on three large constellations - Virgo, Hercules, and Cetus. The size and general position of these constellations[126] are described in Table 3.2. These constellations were chosen due to being approximately the same size, for containing a significant number of galaxies in both datasets, and for their positions relative to each other. Virgo and Hercules are located close together, being in quadrant 3 ($12h \leq \alpha < 18h$) to the south and north of the celestial equator, respectively, while Cetus is located almost exactly opposite Virgo in quadrant 1 ($0h \leq \alpha < 6h$).

The size of these regions limits the number of galaxies available to train and test our machine learning models. The number of galaxies of each morphological type within the selected constellation areas is described in Table 3.3. To avoid over-representing any particular region, we select an equal number of galaxies of each type within each constellation - 6,000 of each from the SDSS and 18,000 of each from Pan-STARRS, totaling 36,000 and 108,000 samples used in the experiment, respectively.

Table 3.2: The size and position of the constellations selected for analysis. Right ascension (RA) and declination (Dec) are given as midpoints of the constellation boundaries.

Constellation	Sky Area		Position		
	deg^2	Percentage	RA $(h\ m)$	Dec $(^\circ\ m)$	Quadrant
Virgo	1294	3.14%	+13 24.39	-04 09.51	SQ3
Hercules	1225	2.97%	+17 23.16	+27 29.93	NQ3
Cetus	1231	2.99%	+01 40.10	-07 10.76	SQ1

Table 3.3: The number of samples for each type of galaxy within each sky region available in either dataset.

		Virgo	Hercules	Cetus
SDSS	Elliptical	8,527	7,463	6,759
	Spiral	8,295	6,675	7,016
	Total	16,822	14,138	13,775
Pan-STARRS	Elliptical	43,217	21,602	38,227
	Spiral	34,094	18,360	29,237
	Total	77,311	39,962	67,464

To compare the generalization capability of the neural network when trained on galaxies from different areas of the sky, we must reserve a large number of samples for evaluation. After deciding on a neural network architecture, we search for a small but substantial training set size by training the network from scratch with an increasingly large subset of the data. A test set of $2 \cdot 10^4$ galaxies is reserved for evaluation, and the neural network is trained five times per trial with a random selection of galaxies (disjoint with the test set) each time. The average accuracy for each trial is recorded and we choose a training set size past the "knee" of the curve, i.e. a point where expanding the training set only marginally improves the accuracy.

3.4.1 Annotation of the SDSS with training sets from different sky regions

Table 3.4: Confusion matrices of Hercules galaxy morphology predictions from two models with training data from different regions. Rows represent true labels and columns the predicted labels.

	Elliptical	Spiral
Elliptical	4929	71
Spiral	147	4853

(a) Homogeneous annotations for the Hercules test set.

	Elliptical	Spiral
Elliptical	4798	202
Spiral	48	4952

(b) Annotations for the same test set but with elliptical galaxies from Virgo used during training.

Performing the network architecture selection process described in Section 3.3 on the SDSS dataset resulted in a four-layer model with 64, 48, 32, and 32 artificial neurons, respectively. This neural network contains a total of $7,154$ trainable parameters. Figure 3.3 displays the loss while training the model, showing that 100 epochs is adequate for the training to converge without overfitting. Figure 3.4 shows how the accuracy of this model increases with the size of the training set. We chose to use 2,000 samples (1,000 of each morphological class) for training, allowing us to reserve a total of 10,000 samples per region for SDSS evaluation and comparison.

To determine if the area of the sky that the training data are selected from induces bias in this model, we train neural networks with the same architecture from scratch using different combinations of training data from each constellation. For example, one model is trained with data taken only from the area of Virgo, while another is trained with elliptical galaxies from Virgo and spiral galaxies from Cetus, and so on. The resulting nine models are then evaluated by predicting class labels for the reserved test set from each region. Table 3.5 shows the accuracy of these models' predictions.

Small variations in model accuracy are expected due to input differences and randomness in the optimization step. We investigate if these variations are significant by applying the binomial test as described in Section 3.3. In this experiment, we have three sets of homogeneous annotations - one for each constellation. Table 3.4 compares the homogeneous

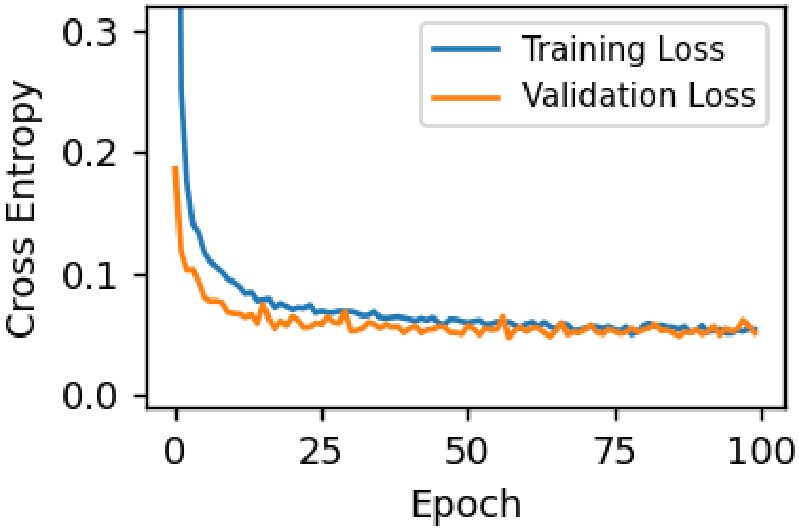


Figure 3.3: Training and validation loss of the neural network architecture with the best performance for the SDSS.

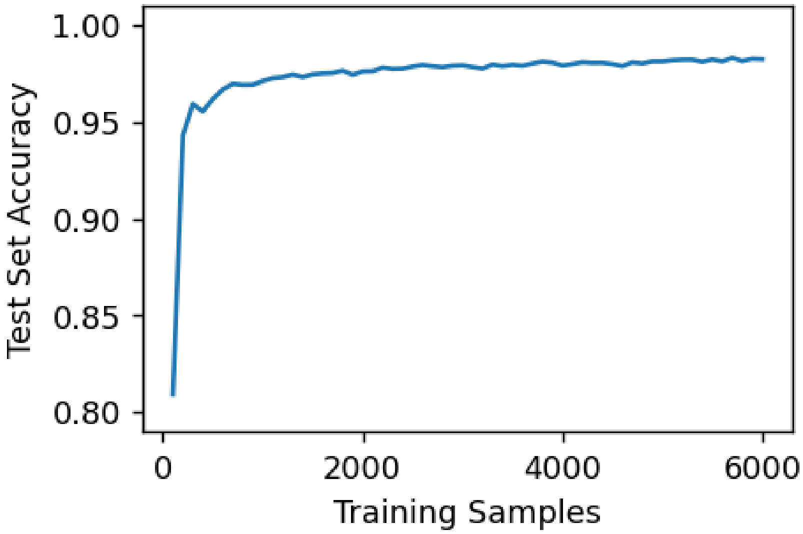


Figure 3.4: Average test-set accuracy of the SDSS model with different training set sizes.

Table 3.5: Prediction accuracy of SDSS models by training set region and test set region.

Training Region		Evaluation Region		
Ellipticals	Spirals	Virgo	Hercules	Cetus
Virgo	Virgo	0.968	0.976	0.979
Virgo	Hercules	0.970	0.975	0.980
Virgo	Cetus	0.966	0.976	0.977
Hercules	Virgo	0.972	0.978	0.981
Hercules	Hercules	0.966	0.978	0.975
Hercules	Cetus	0.961	0.971	0.972
Cetus	Virgo	0.971	0.975	0.980
Cetus	Hercules	0.968	0.971	0.977
Cetus	Cetus	0.973	0.972	0.981

annotations for Hercules to the annotations predicted by the model trained with elliptical galaxies from Virgo and spiral galaxies from Hercules. The expected probability of a correct elliptical label is 0.493, derived from the number of true elliptical labels in Table 3.4a and the size of the test set $(10,000)$. According to the binomial distribution, the probability that the second model correctly labels $4,798$ elliptical galaxies or less in the same test set (see Table 3.4b) is $8.79 \cdot 10^{-3}$, and therefore statistically significant. This suggests that the null hypothesis - that the area of the sky the training data are selected from should not significantly change the distribution of the predicted classes - is unlikely to be true.

Table 3.6 lists the two-tailed p-values from applying the binomial test to each pair of homogeneous annotations and non-homogeneous annotations for the same test set. Statistically significant values ($p < 0.05$) are highlighted in bold. Although most of these values suggest an insignificant variance, an unlucky researcher may "discover" a non-existent anisotropy in galaxy distribution, especially since each of these models achieved a satisfyingly high classification accuracy.

3.4.2 Pan-STARRS by Sky Region

In this Section we repeat the experiment performed in Section 3.4.1 using the Pan-STARRS dataset. In this case, the most successful neural network architecture consisted of two layers with 64 and 32 artificial neurons, respectively, and contains $3,490$ trainable parameters.

Table 3.6: SDSS binomial test p-values. Statistical significance ($p < 0.05$) is indicated in bold.

Training Region		Evaluation Region					
		Virgo		Hercules		Cetus	
Ellipticals	Spirals	Elliptical	Spiral	Elliptical	Spiral	Elliptical	Spiral
Virgo	Virgo	—	—	$9.1 \cdot 10^{-2}$	$1.93 \cdot 10^{-1}$	$5.75 \cdot 10^{-1}$	$3.03 \cdot 10^{-1}$
Virgo	Hercules	$7.11 \cdot 10^{-1}$	$4.96 \cdot 10^{-1}$	$\mathbf{8.79 \cdot 10^{-3}}$	$\mathbf{4.87 \cdot 10^{-2}}$	$9.52 \cdot 10^{-1}$	$6.97 \cdot 10^{-1}$
Virgo	Cetus	$4.01 \cdot 10^{-1}$	$2.42 \cdot 10^{-1}$	$4.29 \cdot 10^{-1}$	$6.89 \cdot 10^{-1}$	$1.87 \cdot 10^{-1}$	$\mathbf{2.78 \cdot 10^{-2}}$
Hercules	Virgo	$6.89 \cdot 10^{-1}$	$7.34 \cdot 10^{-1}$	$2.5 \cdot 10^{-1}$	$2.38 \cdot 10^{-1}$	$3.9 \cdot 10^{-1}$	$3.95 \cdot 10^{-1}$
Hercules	Hercules	$1.8 \cdot 10^{-1}$	$8.18 \cdot 10^{-2}$	—	—	$1.19 \cdot 10^{-1}$	$\mathbf{3.73 \cdot 10^{-3}}$
Hercules	Cetus	$2.71 \cdot 10^{-1}$	$\mathbf{1.04 \cdot 10^{-2}}$	$6.53 \cdot 10^{-1}$	$3.32 \cdot 10^{-1}$	$2.0 \cdot 10^{-1}$	$\mathbf{1.37 \cdot 10^{-3}}$
Cetus	Virgo	$7.79 \cdot 10^{-1}$	$7.19 \cdot 10^{-1}$	$\mathbf{3.94 \cdot 10^{-2}}$	$1.8 \cdot 10^{-1}$	$5.22 \cdot 10^{-1}$	$3.42 \cdot 10^{-1}$
Cetus	Hercules	$4.18 \cdot 10^{-1}$	$4.71 \cdot 10^{-1}$	$\mathbf{7.24 \cdot 10^{-4}}$	$6.14 \cdot 10^{-2}$	$8.49 \cdot 10^{-1}$	$4.9 \cdot 10^{-1}$
Cetus	Cetus	$6.67 \cdot 10^{-1}$	$1.56 \cdot 10^{-1}$	$\mathbf{5.81 \cdot 10^{-4}}$	$\mathbf{2.99 \cdot 10^{-2}}$	—	—

This network's training loss is shown in Figure 3.5, and a similar pattern of convergence is observed as compared with the SDSS model. Figure 3.6 shows how the model's accuracy increases with the number of training samples. We choose to use $6,000$ galaxies in our training set, leaving $30,000$ galaxies for each region's test set. Both of these subsets contain an equal number of spiral and elliptical galaxies.

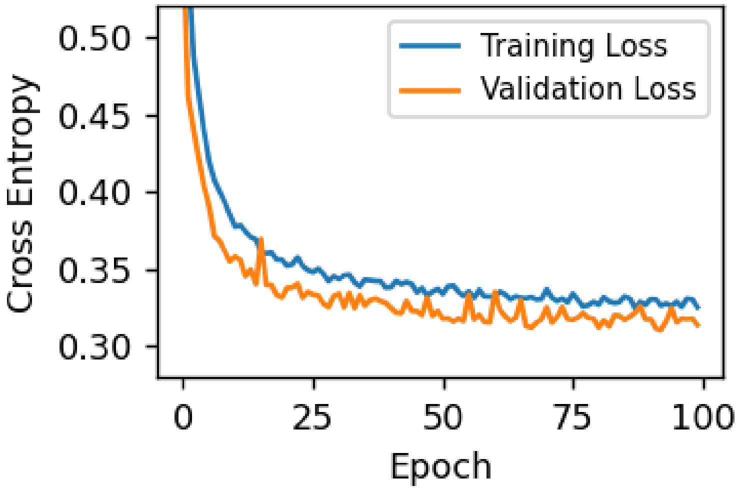


Figure 3.5: Training and validation loss of the winning neural network architecture for Pan-STARRS.

As with the SDSS, we train nine neural networks with the same architecture with training data from each combination of constellation areas. The accuracy of these models is evaluated on the reserved test set from each constellation, and these results are shown in

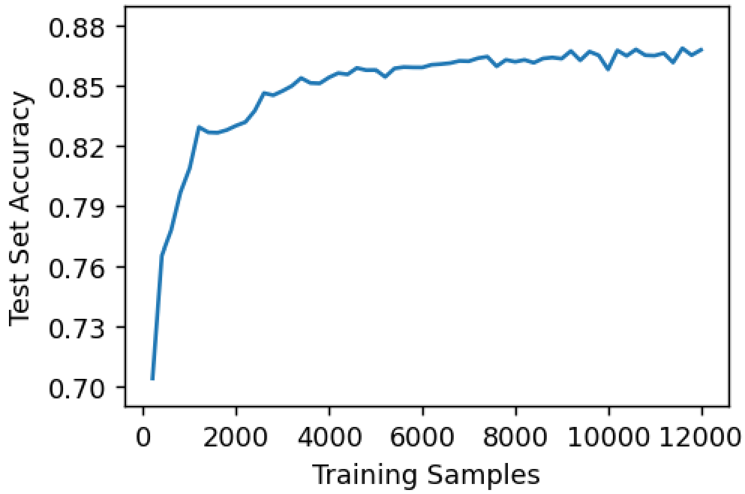


Figure 3.6: Average test-set accuracy of the Pan-STARRS model with different training set sizes.

Table 3.7. Despite being trained with three times as many samples as the SDSS models, these models achieve consistently lower accuracy overall. This may be expected, however, as the magnitude distribution of these galaxies (discussed in Section 3.2) suggests a much more difficult classification task. Regardless, the level of accuracy reached by these models may be acceptable in some real-world analyses.

Table 3.7: Prediction accuracy of Pan-STARRS models by training set region and test set region.

Training Region		Evaluation Region		
Ellipticals	Spirals	Virgo	Hercules	Cetus
Virgo	Virgo	0.861	0.841	0.865
Virgo	Hercules	0.841	0.841	0.841
Virgo	Cetus	0.849	0.821	0.859
Hercules	Virgo	0.836	0.805	0.851
Hercules	Hercules	0.845	0.824	0.840
Hercules	Cetus	0.826	0.789	0.845
Cetus	Virgo	0.859	0.848	0.866
Cetus	Hercules	0.850	0.838	0.843
Cetus	Cetus	0.847	0.825	0.861

The binomial test comparison process described in Section 3.3 is repeated for the Pan-STARRS models, and the p-values are listed in Table 3.8. Compared to the SDSS results, the Pan-STARRS models show a much larger degree of variance in how morphological labels are assigned. While some of this variance can be explained by the relatively lower accuracy of the models, one may expect the error to remain roughly proportionate with respect to class distribution.

Table 3.8: Pan-STARRS binomial test p-values. Statistical significance ($p < 0.05$) is indicated in bold.

		Evaluation Region					
Training Region		Virgo		Hercules		Cetus	
Ellipticals	Spirals	Elliptical	Spiral	Elliptical	Spiral	Elliptical	Spiral
Virgo	Virgo	—	—	**$6.4 \cdot 10^{-34}$**	**$1.4 \cdot 10^{-76}$**	$2.2 \cdot 10^{-1}$	$8.0 \cdot 10^{-1}$
Virgo	Hercules	**$1.7 \cdot 10^{-16}$**	**$1.8 \cdot 10^{-54}$**	**$4.6 \cdot 10^{-25}$**	**$5.4 \cdot 10^{-64}$**	**$2.5 \cdot 10^{-13}$**	**$3.4 \cdot 10^{-48}$**
Virgo	Cetus	**$1.2 \cdot 10^{-13}$**	**$1.5 \cdot 10^{-32}$**	**$1.7 \cdot 10^{-5}$**	**$5.5 \cdot 10^{-4}$**	**$1.1 \cdot 10^{-6}$**	**$2.8 \cdot 10^{-8}$**
Hercules	Virgo	**$2.2 \cdot 10^{-3}$**	**$2.0 \cdot 10^{-8}$**	$1.5 \cdot 10^{-1}$	**$5.0 \cdot 10^{-8}$**	$2.9 \cdot 10^{-1}$	**$5.9 \cdot 10^{-6}$**
Hercules	Hercules	**$1.0 \cdot 10^{-19}$**	**$4.6 \cdot 10^{-50}$**	—	—	**$2.9 \cdot 10^{-23}$**	**$2.4 \cdot 10^{-69}$**
Hercules	Cetus	**$6.2 \cdot 10^{-5}$**	**$1.1 \cdot 10^{-16}$**	$1.3 \cdot 10^{-1}$	**$2.2 \cdot 10^{-28}$**	**$3.0 \cdot 10^{-2}$**	**$1.0 \cdot 10^{-3}$**
Cetus	Virgo	**$5.0 \cdot 10^{-3}$**	**$3.1 \cdot 10^{-2}$**	**$4.6 \cdot 10^{-23}$**	**$8.4 \cdot 10^{-79}$**	**$1.8 \cdot 10^{-2}$**	$5.2 \cdot 10^{-1}$
Cetus	Hercules	**$4.9 \cdot 10^{-6}$**	**$2.6 \cdot 10^{-17}$**	**$7.1 \cdot 10^{-43}$**	**$8.7 \cdot 10^{-83}$**	**$2.3 \cdot 10^{-12}$**	**$3.3 \cdot 10^{-41}$**
Cetus	Cetus	**$3.7 \cdot 10^{-3}$**	**$3.9 \cdot 10^{-2}$**	**$4.6 \cdot 10^{-23}$**	**$5.6 \cdot 10^{-26}$**	—	—

Consider from this experiment the model P_{CH}, trained with elliptical galaxies from Cetus and spiral galaxies from Hercules, and the homogeneous annotations for Hercules, produced by model P_{HH} (trained only with galaxies from Hercules). The predicted class labels for the Hercules test set by P_{CH} have very similar accuracy to the homogeneous annotations, differing by only 1.4 percentage points. However, the morphological distribution predicted by these models is extremely different, which is reflected by the values in Table 3.8 ($7.1 \cdot 10^{-43}$ and $8.7 \cdot 10^{-83}$). From the "perspective" of model P_{HH}, about 37% of the galaxies in the Hercules test set are spiral. The only methodological difference in model P_{CH} is that the elliptical galaxies used to train it are sampled from Cetus rather than Hercules, yet it sees the same test set as being 46% spiral.

3.5 Generalizing to Different Sky Surveys

On the surface, training a machine learning model on a mature sky survey seems like a straightforward way to begin rapid analysis of large datasets produced by more modern telescopes. However, differences in technology, survey methodology, or the location of the

Table 3.9: Unsupervised clustering using a 2-component Gaussian mixture model.

	Cluster 1	Cluster 2
Elliptical	567,180	108,221
Spiral	222,830	340,713

(a) Cluster assignment compared to morphological class.

	Cluster 1	Cluster 2
Pan-STARRS	729,461	262,057
SDSS	60,549	186,877

(b) Cluster assignment compared to source survey.

telescope can make it very difficult to apply one model to multiple surveys. Deep neural networks are sensitive to changes in how data are collected, processed, or formatted, often leading to generalization failure if those same conditions are not met for new observations or the model is not adjusted to compensate. One simple example of this is a difference in the resolution of imagery.

The photometric filters used by telescopes determine the wavelengths of light the telescope is able to detect. Astronomical objects can have drastically different appearances when observed at different wavelengths for many reasons, including light emission/reflection properties, redshift, or being obscured by cosmic dust. Machine learning models are very unlikely to be able to generalize to other sky surveys that do not share photometric filters (e.g. a visible light telescope and a mid-infrared telescope), even if they are observing the same objects.

In this experiment, we investigate the ability of a deep neural network to predict galaxy morphology from photometric observations from both the SDSS and Pan-STARRS. We limit our feature vector to only the shared fields in Table 3.1 (i.e. we ignore the u band in the SDSS and the y band in Pan-STARRS). This results in a vector of 17 inputs for the machine learning models. The combined dataset contains $1,238,944$ samples; however, due to the overlap in the areas surveyed, this is not necessarily the number of unique objects.

Before constructing the neural network, we investigate the similarity of the samples in the combined dataset with unsupervised clustering. We apply a 2-component Gaussian mixture model to the whole dataset and compare the composition of the two resulting clusters. Table 3.9 shows how the data are assigned to each cluster with respect to morphological class and the source survey. The quality of the cluster assignments can be compared using the Matthews correlation coefficient (MCC), which ranges from -1 (inverse prediction) to 1 (perfect prediction) and handles imbalanced classes well. These clusters have an MCC of 0.46 when evaluated with morphological class and an MCC of 0.41 when evaluated with the source survey. In other words, the separability of these samples by morphology or by which survey they belong to is close to the same.

3.5.1 Annotating the SDSS and Pan-STARRS

The neural network architecture for this experiment is determined with the same process as in Section 3.4 and is trained with an approximately even mixture of samples from both surveys. The network with the highest test-set accuracy used three layers with 64, 64, and 48 artificial neurons, respectively, and contains $8,350$ trainable parameters. Figure 3.7 shows the training loss of this model, which displays slightly more volatility relative to the single-survey models but still approaches convergence fairly well.

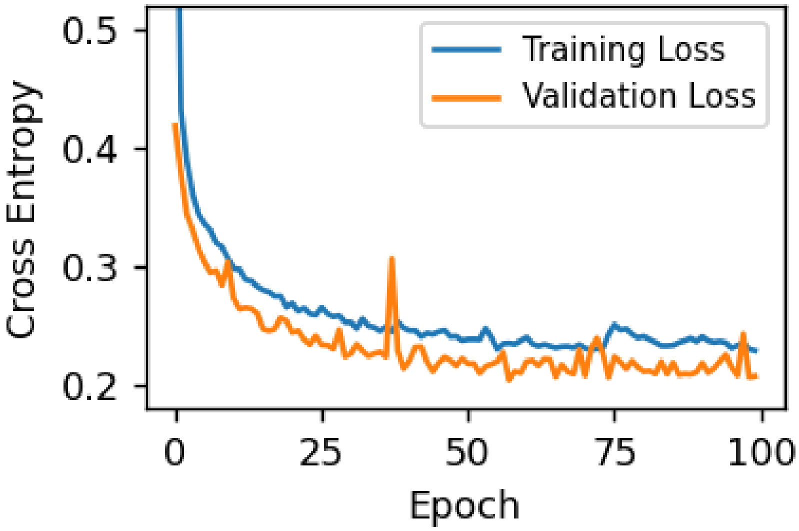


Figure 3.7: Training and validation loss of the neural network architecture with the best performance when trained with a mixture of data from the SDSS and Pan-STARRS.

We select $1 \cdot 10^5$ galaxies of each morphological class from both surveys, divided proportionally into 80% for training and 20% for testing. Spiral galaxies from one survey and elliptical galaxies from another, totalling $1.6 \cdot 10^4$ samples, are used to train four neural networks with the same architecture. Table 3.10 lists the classification accuracy for the test sets from each survey, which contain $4 \cdot 10^4$ galaxies each. As expected, the homogeneous annotations (i.e. when the test set comes from the same survey as all training samples) have comparable accuracy to the single-survey neural networks in Sections 3.4.1 and 3.4.2.

Table 3.10: Test-set classification accuracy by models trained on different combinations of data from the SDSS and Pan-STARRS (PS1).

Training Source		Evaluation Survey	
Ellipticals	Spirals	SDSS	Pan-STARRS
SDSS	SDSS	0.990	0.559
SDSS	PS1	0.525	0.506
PS1	SDSS	0.869	0.576
PS1	PS1	0.840	0.868

The model trained only on SDSS samples fails to generalize to the Pan-STARRS test set, classifying 94% of those galaxies as elliptical. On the other hand, the model trained only with Pan-STARRS samples achieves fairly high accuracy on the SDSS test set, albeit significantly lower than the SDSS-trained model. When trained with elliptical galaxies from the SDSS and spirals from Pan-STARRS, the model annotates 97% of the SDSS test set as elliptical and 99% of the Pan-STARRS test set as spiral, indicating that this model learned to distinguish samples based on which survey they originated from rather than by morphology.

These results strongly suggest survey-level differences in the data, such as the distribution of magnitudes or the sensitivity of each telescope's photometric filters. Neural networks are able to learn to distinguish samples based on these "features" in addition to (or sometimes instead of) the aspects of the data one would like to teach the model. In some tasks these differences are too subtle or obscure for a human to notice and correct for. Additionally, when differences are noticed, adapting the data or the machine learning model to correct for it may be non-trivial.

3.6 Conclusions

The information era has changed astronomy research by enabling data-driven research with the use of very large astronomical databases. Instruments generating vast pipelines of astronomical data reinforce the need for automatic methods that can annotate the data in a manner that makes it suitable for analysis. Due to the size of the data, discoveries cannot be made practically by manual analysis alone, and therefore automating the annotation of the data is required.

One of the immediate solutions to the automatic annotation of data is the use of machine learning. Machine learning can handle the annotation of complex data with high accuracy, yet without the need to design a specific model-driven solution to each annotation task. While machine learning does not require a labor-intensive step of defining a model, it provides annotation accuracy that very often exceeds the accuracy of a manually-crafted, knowledge-driven solution. The ability to annotate complex data with high accuracy makes machine learning an effective solution to the problem of annotating very large astronomical databases, and it has been used for multiple tasks in this field.

While machine learning provides many advantages, it should also be analyzed for its potential downsides. Here we analyze the potential bias driven by the source of the training samples. Experimental results show that despite using a training set from the same astronomical survey, the distribution of the locations of the samples used for training affects the annotations. That is, the exact same set of galaxies is annotated in a different way based on where in the sky the training data are sampled from.

To notice that difference, the user of the data products needs to be familiar with the details of how the training samples are collected and distributed. Therefore, a user of an astronomical data product created using machine learning may not be aware of such bias, and might therefore reach conclusions that are influenced by the machine learning bias rather than by the real distribution of astronomical objects in the sky. Without fully understanding all details of the construction of the training set, it is difficult to know whether the statistical distribution of the annotations indeed reflects the real sky.

Biases in data collected by digital sky surveys, including noise in measurements, are a known characteristic of these powerful instruments. For instance, different parts of the sky might have slight but statistically significant differences in the limiting magnitude. By using machine learning the bias becomes more complex and less intuitive. As an example, it can be assumed that parts of the sky that provide better limiting magnitude can also provide better imaging, and therefore it can be expected that a higher number of galaxies will be annotated as spiral due to the increased visibility of spiral arms. With the bias driven by machine learning, if the training set that was used to annotate the data had a higher number of galaxies from the part of the sky with lower limiting magnitude, the annotation might lead to a catalog in which more spiral galaxies are identified in the parts of the sky with lower limiting magnitude, which might seem puzzling to an unsuspecting observer.

We also demonstrate the challenges of using a machine learning model trained with samples from one survey to generate annotations for another, even when using the same photometric features. Many aspects of a sky survey can create subtle patterns in the data that are practically invisible to humans. Differences in the sensitivity of photometric filters, atmospheric conditions at telescope sites, data processing pipelines, and the type of objects the survey intends to observe can create biases in machine learning models. These biases may be insignificant when use of the model is limited to the same survey, but may have large effects if the model is applied to other datasets.

While machine learning provides a useful solution to the annotation of very large astronomical databases, it also has several downsides. Many researchers without a background in deep learning may not be aware of these downsides, unknowingly introducing biases into their own analyses when using data products created with machine learning. Since biases are difficult to identify and profile in annotations made by machine learning systems, data products prepared with such methods should be used with caution.

Chapter 4

Experiments with Vision Transformers for Fine-Grained Galaxy Morphology Classification

4.1 Introduction

The study of astronomical objects generates a vast amount of data, which requires sophisticated data analysis techniques to extract meaningful information. Over the past few decades, astronomers have employed a range of data-driven approaches, from early statistical methods to more recent deep learning techniques, to analyze and classify astronomical data. With the increasing availability of large datasets, machine learning techniques have become increasingly important in astronomy[19;23;31;85;105;127–131].

In recent years, transformer-based models have emerged as a powerful tool in natural language processing (NLP). Transformers were first introduced in the seminal paper by Vaswani et al.[132], which introduced the concept of self-attention as a mechanism for processing sequences of variable length. Since then, transformers have achieved state-of-the-art results in various NLP tasks, such as language modeling, question answering, and machine translation[133–137].

The success of transformers in NLP has led researchers to explore their potential in other domains, including computer vision. Vision transformers (ViTs) were introduced by Dosovitskiy et al.[138] as a novel architecture for image classification, which has shown promising results in various computer vision tasks[139–141].

The classification of galaxy morphology is a fundamental task in astronomy, which aims to categorize galaxies based on their structural properties, such as shape, spiral arms, and the presence of bars or rings. Galaxy morphology classification has been traditionally performed using hand-crafted features and traditional machine learning algorithms, such as support vector machines (SVMs) and random forests[18;32;36;108]. However, with the availability of large-scale datasets, deep learning approaches have also been applied to this task, with CNNs being the most commonly used architecture[85;106;142;143].

In this paper, we investigate the effectiveness of ViTs for the task of fine-grained galaxy morphology classification. Although our experiments do not show a significant improvement over recent CNN-based models, we believe that our results provide valuable insights into the challenges and opportunities in applying transformers to image analysis tasks in astronomy.

4.2 Galaxy Zoo

The Galaxy Zoo project is a citizen science project that invites members of the public to classify images of galaxies taken from various sky surveys. The goal of the project is to improve our understanding of the properties and evolution of galaxies by collecting a large and diverse dataset of galaxy classifications. The project was launched in 2007 and has since grown to include multiple surveys and classification tasks. Our experiment uses four different Galaxy Zoo projects, described in this section.

Galaxy Zoo 2 (GZ2)[144;145] includes images from the Sloan Digital Sky Survey (SDSS), a large-scale survey that has imaged and spectroscopically measured several million galaxies. The images in GZ2 are from the SDSS DR7 and cover a sky area of approximately 8,000 square degrees. The dataset includes labels for morphological classifications for a subset of the SDSS (approximately 243,000 galaxies). Users were asked to classify each galaxy according to its shape, such as whether it has a smooth or clumpy appearance, whether it has a central bulge, and whether it has spiral arms.

Galaxy Zoo Hubble (GZH)[146] includes images from the Hubble Space Telescope (HST), a space-based telescope that observes in the ultraviolet, visible, and near-infrared regions of the electromagnetic spectrum. The images in GZH come from the Advanced Camera for Surveys (ACS) instrument aboard the telescope. The dataset includes labels for multiple tasks related to galaxy morphology, including classifications of disk structure, bar presence, and spiral arm winding. GZH includes every task present in GZ2 but added tasks related to ”clumps”, such as the number, arrangement, and symmetry if any are present.

Galaxy Zoo CANDELS (GZC)[147] includes images from the Cosmic Assembly Near-infrared Deep Extragalactic Legacy Survey (CANDELS), a multi-wavelength survey that uses HST to observe galaxies in the distant universe. CANDELS images were taken with both the ACS and Wide Field Camera 3 (WFC3) instruments on the Hubble Space Telescope, focusing on rest-frame optical wavelengths of galaxies at redshifts of $1 < z < 3$. The tasks for GZC are almost the same as those for GZH, but replaces a broad question about ”odd” features with a more specific task to identify galactic mergers and tidal debris.

Galaxy Zoo DECaLS (GZD)[142] includes images from the Dark Energy Camera Legacy Survey (DECaLS). This survey uses the Dark Energy Camera (DECam) on the Blanco 4m telescope, located at the Cerro Tololo Inter-American Observatory in Chile, to observe galaxies in the optical and near-infrared wavelengths. Our work uses the tasks and user responses from the GZD-5 campaign, which has questions similar to GZ2 but is designed to more accurately detect mergers and galaxies with weak bars.

Table 4.1: The number of labeled images used from each Galaxy Zoo project.

Project	GZ2	GZH	GZC	GZD	Total
Number of Images	243,434	100,788	49,552	249,581	643,355
Percentage of Dataset	37.8%	15.7%	7.7%	38.8%	100%

In our experiments, we only use entries from the Galaxy Zoo datasets with at least 3 volunteer classifications. Table 4.1 shows the size of our combined dataset and the number of images from each Galaxy Zoo project. A summary of the tasks and responses in each of the projects is shown in Tables 4.2 and 4.3. Example images from each project are included in Table 4.4.

4.3 Experiment Design

Our experiment is based on prior work in automated morphological classification of galaxies using deep learning, namely Zoobot[148], released alongside Galaxy Zoo DECaLS[142]. Zoobot is a Bayesian deep learning model implemented using the EfficientNet-B0[149] convolutional neural network architecture. It is designed to predict the posterior probabilities of each answer to all tasks in GZD, as well as producing a representation vector that can be used to search for galaxies based on their similarity to a candidate galaxy.

The creators of Zoobot leverage the information shared between tasks to predict the responses to all GZD tasks with one model - in their words, ”intuitively, knowing how to recognize spiral arms can also help you count them.” We attempt to extend this intuition to the shared information between similar or identical tasks in each Galaxy Zoo project to train one model that can classify galaxies from different sky surveys.

Tables 4.2 and 4.3 show the considerable overlap in the information available in the labels for each dataset, as well as several tasks present in some projects but not others. Ideally, training one model on all four Galaxy Zoo projects should both reinforce the learning of shared tasks and teach the model how to answer tasks for galaxies whose projects did not contain those tasks (e.g. detecting details about clumps in GZ2 and GZD images). The ability to make predictions in multiple-classification problems where labels are incomplete[150;151] or even completely unobserved[152;153] is an important but under-explored topic in machine learning. We believe that this combination of partially-overlapping datasets provides for a unique and challenging approach to this problem.

One of our goals is to address the generalization challenges described in Section 3.5. The Galaxy Zoo projects we use contain imagery collected in a variety of wavelengths with a mixture of ground- and space-based telescopes. By training our model on this mixture of image domains with a unified set of labels, we hypothesize that the model will be encouraged

Table 4.2: A list of tasks and responses in the four Galaxy Zoo projects used in the experiment. In practice, each task included example diagrams to assist the user in choosing their response.

Task	**Response**	**GZ2**	**GZH**	**GZC**	**GZD**
Smooth or Featured?	Smooth	✓	✓	✓	✓
	Features or Disk	✓	✓	✓	✓
	Star or Artifact	✓	✓	✓	✓
How Rounded is the Galaxy?	Completely	✓	✓	✓	✓
	In Between	✓	✓	✓	✓
	Cigar Shaped	✓	✓	✓	✓
Disk Viewed Edge-On?	Yes	✓	✓	✓	✓
	No	✓	✓	✓	✓
Does the Galaxy Have a Bar?	Yes	✓	✓	✓	
	Strong				✓
	Weak				✓
	No	✓	✓	✓	✓
Does the Galaxy Have Spiral Arms?	Yes	✓	✓	✓	✓
	No	✓	✓	✓	✓
How Tightly Wound are the Arms?	Tight	✓	✓	✓	✓
	Medium	✓	✓	✓	✓
	Loose	✓	✓	✓	✓
How Many Arms?	1	✓	✓	✓	✓
	2	✓	✓	✓	✓
	3	✓	✓	✓	✓
	4	✓	✓	✓	✓
	More Than 4	✓	✓	✓	✓
	Can't Tell	✓	✓	✓	✓
Bulge in Center (Edge-on)?	Yes			✓	
	No			✓	
How is the Bulge Shaped (Edge-on)?	Rounded	✓	✓	✓	
	Boxy	✓	✓	✓	
	No Bulge	✓	✓	✓	
How Prominent is the Bulge?	No Bulge	✓	✓	✓	✓
	Just Noticeable	✓	✓		✓
	Moderate				✓
	Obvious	✓	✓	✓	✓
	Dominant	✓	✓	✓	✓

Table 4.3: A list of tasks and responses in the four Galaxy Zoo projects used in the experiment (continued).

Task	**Response**	**GZ2**	**GZH**	**GZC**	**GZD**
Is the Galaxy Clumpy?	Yes		✓	✓	
	No		✓	✓	
Is One Clump Brightest?	Yes		✓	✓	
	No		✓	✓	
Is the Brightest Clump Central?	Yes		✓	✓	
	No		✓	✓	
How are the Clumps Arranged?	Line		✓	✓	
	Chain		✓	✓	
	Cluster		✓	✓	
	Spiral		✓	✓	
How Many Clumps?	1		✓	✓	
	2		✓	✓	
	3		✓	✓	
	4		✓	✓	
	More Than 4		✓	✓	
	Can't Tell		✓	✓	
Is the Galaxy Symmetrical?	Yes		✓	✓	
	No		✓	✓	
Clumps Embedded in Larger Object?	Yes		✓	✓	
	No		✓	✓	
Is There Anything Odd?	Yes	✓	✓		
	No	✓	✓		
What is the Odd Feature?	Ring	✓	✓		
	Lens or Arc	✓	✓		
	Disturbance	✓	✓		
	Irregular	✓	✓		
	Other	✓	✓		
	Merger	✓	✓		
	Dust Lane	✓	✓		
Merger or Tidal Debris?	Merging			✓	
	Tidal Debris			✓	
	Both			✓	
	Neither			✓	
Merger or Disturbance?	None				✓
	Minor Disturbance				✓
	Major Disturbance				✓
	Merger				✓

to focus primarily on the visual aspects in each image while putting less emphasis on the details of how the data were collected. Furthermore, some objects are imaged in more than one of the Galaxy Zoo projects, giving the model a different view of the object (perhaps with additional labeled information) beyond what is accomplished with augmentations during training.

4.3.1 Vision Transformer

The Vision Transformer (ViT) is a type of neural network architecture that has gained considerable attention for its ability to achieve state-of-the-art performance on various computer vision tasks, including image classification and object detection. ViT is based on the transformer architecture, which was originally developed for natural language processing (NLP) tasks[132]. The architecture is based on the idea of self-attention, where the input sequence is transformed using a series of attention mechanisms. This enables the model to focus on different parts of the input during processing and to learn which input tokens are more or less relevant to each other token.

The basic architecture of the ViT model consists of two main components: a patch embedding module and a transformer encoder. The patch embedding module is responsible for dividing the input image into a sequence of non-overlapping fixed-size patches, which are then flattened and embedded into a sequence of feature vectors[138]. The projection is learned during training using a linear layer. This process loses the two-dimensional spacial relationship between image patches, so a ”position embedding” vector is calculated based on the input dimensions and added to the embedded input sequence. Finally, one extra token (typically called the CLS token) is prepended to the feature vectors, acting as a sort of internal memory for the transformer encoder network. The resulting sequence of embedded vectors is then passed to the transformer encoder for further processing. The length of these vectors is referred to as the model dimension. Our ViT model contains four separate patch embedding modules - one for each Galaxy Zoo dataset - so that it may more easily translate images from different sources into a coherent set of feature vectors for the encoder.

The transformer encoder consists of a series of N identical transformer blocks, each of which consists of a self-attention layer and a feedforward layer. The self-attention layer computes the attention scores between each pair of feature vectors in the input sequence, allowing the model to learn the relationship between patches in different locations of the input image. Self-attention is implemented using multiple attention ”heads” that each observe a separate portion of the feature vectors, encouraging the model to compartmentalize different details of each input. The feedforward layer is a fully connected multi-layer perceptron (MLP) with one hidden layer, providing a nonlinear transformation of the input feature vectors. Each encoder block includes a skip connection after the self-attention layer and the feedforward layer, and each MLP in the encoder uses dropout[124] at a rate of 20%.

While transformer models usually include layer normalization steps in each encoder block, ours omits these by implementing the T-Fixup[154] method for weight initialization and gradient scaling during the training process. This method also removes the need for learning rate warmup, which is often used to avoid unstable gradients in the early stages of training transformer models. In addition, our ViT model implements sparse mixture-of-experts (MoE)[155;156] with expert choice routing[157] in the feedforward layer. Sparse MoE trades increased training time (via more model parameters) for significantly faster inference time by selectively activating different portions of the network during the forward pass. This is desirable for use with the large, autonomously-collected imagery datasets that have become the status quo in astronomy research.

The output of the transformer encoder is a sequence of transformed feature vectors including the CLS token. Only the CLS token is used as input for the classification head, which typically consists of one fully connected layer and a softmax activation function for predicting class probabilities. Our ViT model contains one classification head that combines the tasks from each of the four Galaxy Zoo projects described in Section 4.2, and each output neuron represents one of the responses for a task. The classification head contains 66 outputs after combining some of the redundant answers present in different tasks. For example, in Table 4.3, the responses identifying a merger in the final two tasks (GZC and GZD) are assigned to the same neuron for the merger response of the GZ2/GZH task for identifying odd features.

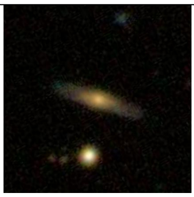
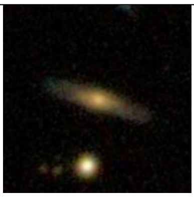
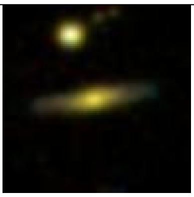
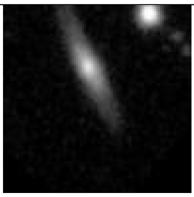
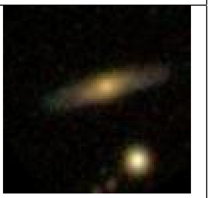
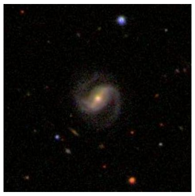
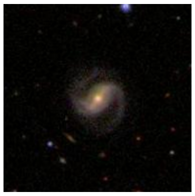
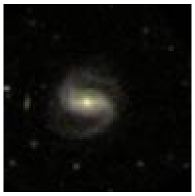
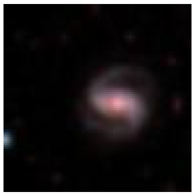
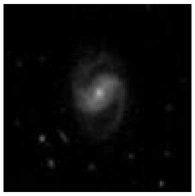
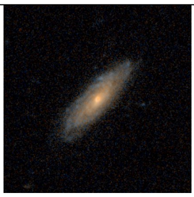
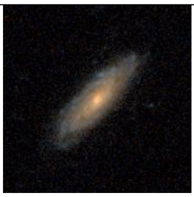
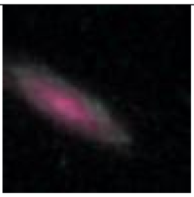
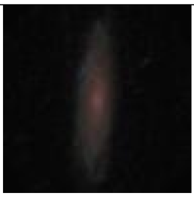
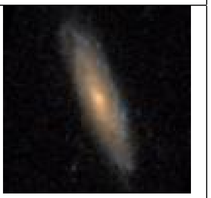
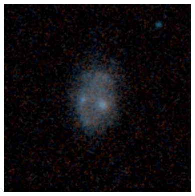
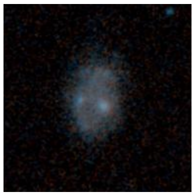
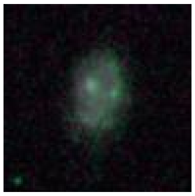
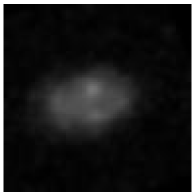
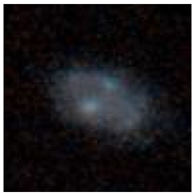
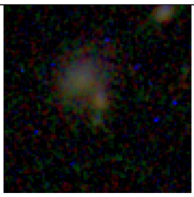
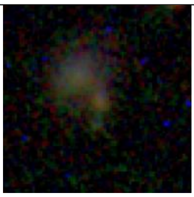
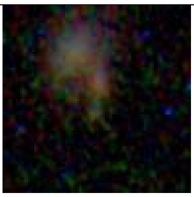
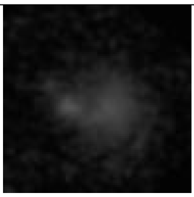
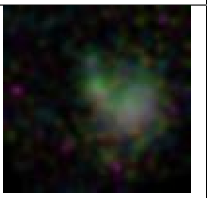
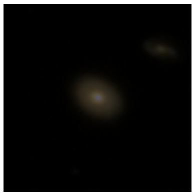
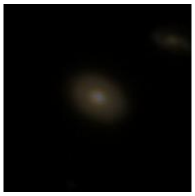
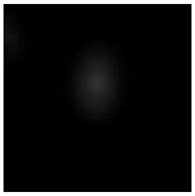
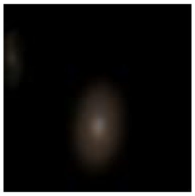
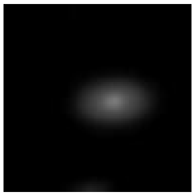
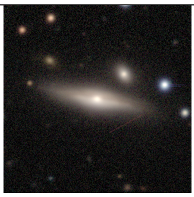
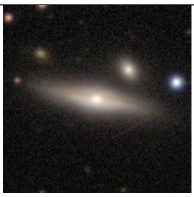
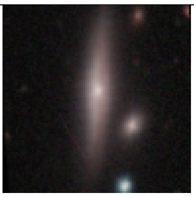
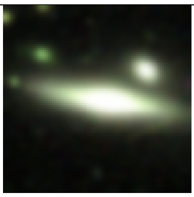
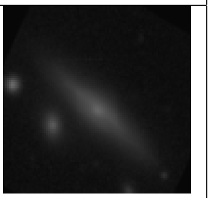
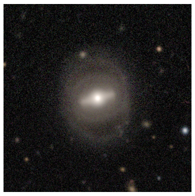
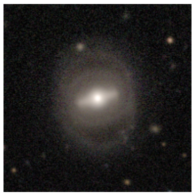
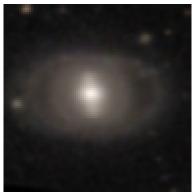
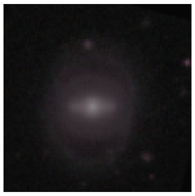
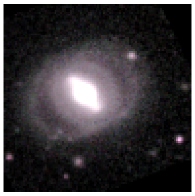
In Bayesian classification, it is common to model the uncertainty in the predictions by using a probability distribution over the class labels. We follow the practice of Walmsley et al.[142] and use the Dirichlet-multinomial distribution, a multivariate extension of the binomial and beta distributions that can be readily applied to problems with more than two classes. For each task, our model predicts the concentration parameter $\vec{\alpha}$ of the Dirichlet distribution with one value per response, which can then be used to sample from the distribution to obtain the predicted answer to the task. This approach enables the model to express the uncertainty in the predictions and to provide a more informative output than a point estimate. The model's loss is computed as the negative log likelihood between the sampled distribution and the votes for each task response from Galaxy Zoo volunteers.

4.3.2 Training Procedure

All images in the dataset are originally 424 px square. Prior to augmentation in the training step, a 360 px square crop is taken from the center of the image which removes mostly empty space and extragalactic objects, and the resulting crop is resized to 192 px square. This version of each image is given directly to the model during the validation and test steps. In the training step, images are augmented using the following process: first, the image is rotated by a random angle between $-180°$ and $180°$. The image is then cropped to between 65% and 95% of its current size at a random off-center point and resized to 96 px square. Third, there is an 80% chance to apply color jitter, randomly changing the brightness, contrast, and saturation by up to 40% and the hue by up to 20%. After this a small amount of Gaussian blur is applied with a 50% probability, and last the image is converted to grayscale with a 30% probability. Table 4.4 shows examples of original images and some possible results of this augmentation process.

The patch embedding modules divide each augmented image into 64 12x12 px patches with 3 color channels and embeds these into a 1024-dimensional feature vector. The transformer encoder has a depth of $N = 8$ blocks. The self-attention layers use 8 attention heads

Table 4.4: Example images from each Galaxy Zoo project with the validation crop and examples of possible augmentations during the training step.

Project	Original	Validation	Example Augmentations		
GZ2					
					
GZH					
					
GZC					
					
GZD					
					

that each process a 128-dimensional slice of the input vectors. Each feedforward layer uses 8 experts which, with expert-choice routing, select up to 25% of the input vectors to process. Each expert's MLP uses a hidden dimension of 1024, matching the model dimension. In total, the model contains approximately 169 million trainable parameters.

The model is trained using a batch size of 256. Images in the batch are grouped based on which Galaxy Zoo project they are from and routed through the appropriate patch embedding module. The entire batch of embedded vectors is passed through the encoder and classification head at once, after which they are again separated into groups to calculate loss. A specific subset of the output neurons are selected corresponding to the group's tasks, determined by a Galaxy Zoo task schema provided to the model at initialization. The total loss for the batch is the sum of the partial loss from each group. Our model is optimized using AdamW[158] (Adam[125] plus weight decay) with AMSGrad[159]. We use an initial learning rate of 10^{-5} and an adaptive learning rate scheduler that reduces the rate by a factor of 0.2 if the validation loss does not improve after three consecutive training epochs.

4.4 Experimental Results

Figure 4.1 shows the loss of our ViT model during training. Several combinations of hyperparameters were tested and all showed similar behavior - loss decreased sharply in the first few epochs of training then plateaued with little to no further improvement. Loss on the validation set remained close to the same value as on the training set, suggesting that the model was able to learn enough to recognize some morphological features in the larger, non-augmented validation images.

In order to observe the effect of training the model with multiple Galaxy Zoo projects simultaneously, we also trained the same model with each of the four datasets individually. The only difference in these experiments is the ViT classification head, which is set to exactly match the task schema of the respective project. Due to the significant difference in the size of each dataset, the models were trained until they had seen 5,000 training batches (all containing 256 images) rather than for a set number of epochs.

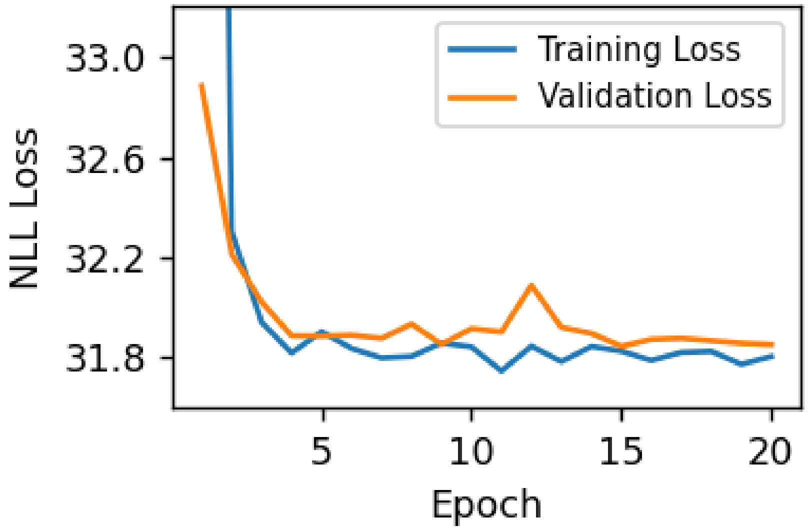


Figure 4.1: Training and validation loss of our ViT on the combined Galaxy Zoo dataset.

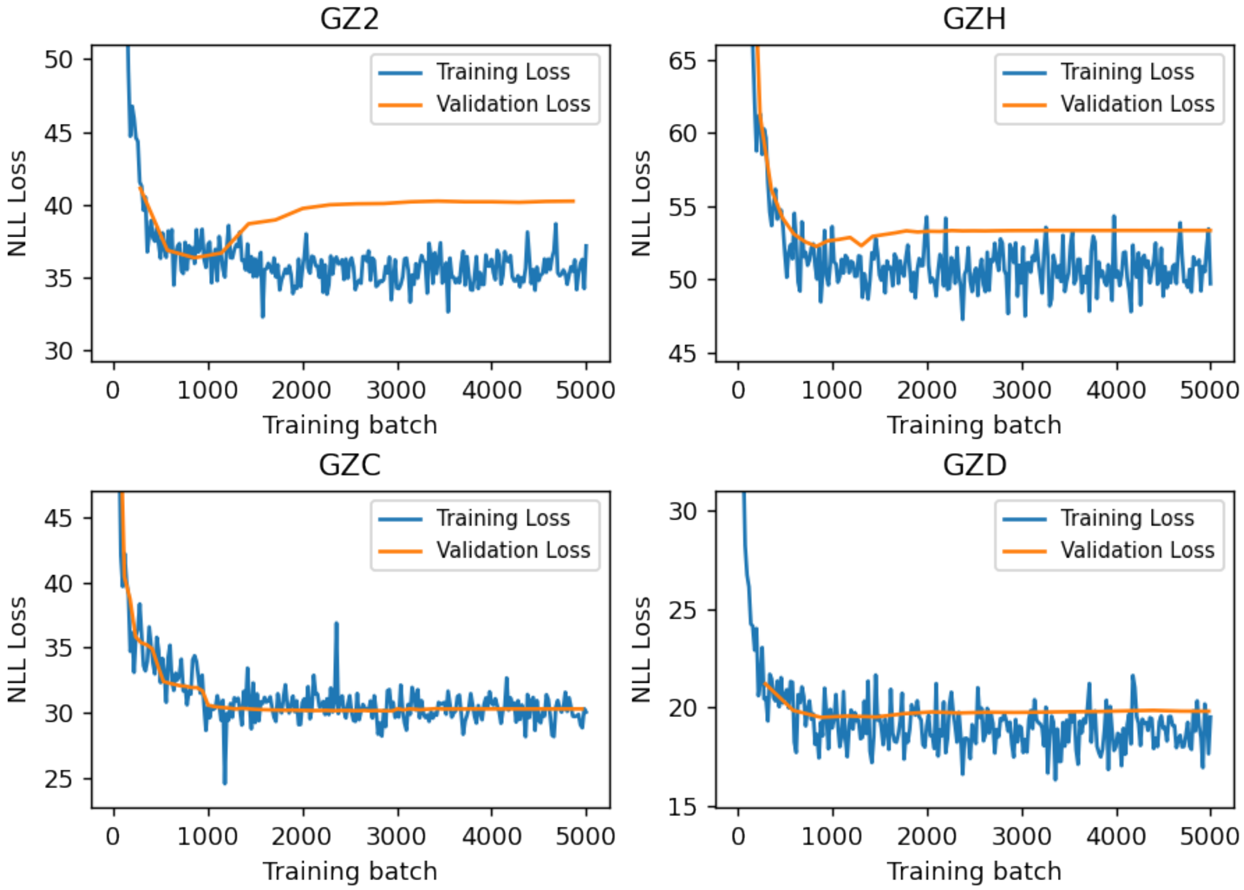


Figure 4.2: Training and validation loss of the ViT when trained on each Galaxy Zoo dataset individually.

Figure 4.2 shows the loss throughout training for each of the four single-dataset models. These display behavior very similar to the combined model, in that the loss quickly levels out an fails to improve significantly. The effect of learning rate decay is apparent here, particularly in the early improvement of the GZC model. Unlike the combined model, the GZ2 model shows obvious signs of overfitting in the validation loss curve. This may indicate that the use of multiple datasets with similar labels acts as a sort of regularization to prevent overfitting; however, more experiments are needed before this can be confirmed.

We evaluate the accuracy of the model as the number of tasks answered correctly out of all tasks in the schema (for each dataset individually), averaged over the test set. Table 4.5 compares the accuracy of the combined model to each of the single-dataset models. It is important to note that the test sets are not identical in each column, as the combined dataset was split into training and testing sets after the combination. However, it can be seen that the combined model performed significantly better on the GZ2 and GZD projects than the models trained on those datasets individually. These two projects have fewer tasks then the others but are the largest datasets in the experiment, containing over 3/4 of the total images in the combined dataset. The smallest dataset, GZC, may have seen its accuracy reduced in the combined model due to its imbalanced representation.

Table 4.5: The accuracy of the combined model on each project's subset of the combined test set, and the accuracy of each individually-trained model on its respective test set.

	GZ2	GZH	GZC	GZD
Accuracy (combined model)	53.5%	50.0%	48.5%	44.4%
Accuracy (individual model)	43.8%	49.3%	50.4%	41.6%

The accuracy of these models in isolation is currently too low for practical use, but the results indicate that the ViT was able to learn morphological details far beyond the level of random guessing. Furthermore, the results strongly suggest that the combination of multiple Galaxy Zoo datasets along with the overlapping morphological labels helped improve the overall performance of the model.

4.5 Conclusions

In this paper, we investigated the potential of vision transformers for the task of fine-grained galaxy morphology classification, using several Galaxy Zoo datasets. Our experiments showed that ViTs did not perform as well as mature CNNs for this task; however, our study provides valuable insights into the challenges and opportunities in applying transformers to morphological classification tasks in astronomy.

Our tentatively optimistic results highlight the importance of careful evaluation and benchmarking when introducing new techniques and models to a field, especially when the existing approaches have already achieved high performance. Nonetheless, we believe that ViTs still have great potential for image analysis in astronomy, and our study provides a starting point for future research exploring the effectiveness of the many existing different ViT architectures for galaxy morphology classification, as well as their applications to other astronomical image analysis tasks.

Several other promising avenues for future work exist that can continue from the results presented in this study. Our technique for learning from different Galaxy Zoo datasets with one model could be applied to neural networks such as Zoobot[142], which have already shown success in morphology classification for individual Galaxy Zoo projects (DECaLS in this case). Furthermore, other work in self-supervised learning for ViTs[140] has shown that the self-attention layers can be used to generate segmentation maps without training the model specifically to do so. Hypothetically, a ViT that was able to correctly answer certain tasks for spiral galaxies (such as the number of arms and how tightly they are wound) could be used to identify the location of the spiral arms in the galaxy images. Another promising idea is to incorporate multimodality[160] into the model. Astronomical images often have significant additional data, such as photometry and redshift measurements, associated with them in public databases. Incorporating this extra information into the patch embedding modules of the ViT would likely improve the performance of the model.

Our work contributes to the ongoing efforts to apply state-of-the-art machine learning techniques to astronomical data analysis, and we hope that our results will encourage further exploration and innovation in this field. Ultimately, the development of more advanced and effective machine learning algorithms will enable astronomers to analyze and interpret the ever-increasing amount of data generated by astronomical surveys, leading to new discoveries and insights into the nature of the universe.

Bibliography

[1] SG Djorgovski, RJ Brunner, AA Mahabal, SC Odewahn, RR de Carvalho, RR Gal, P Stolorz, R Granat, D Curkendall, J Jacob, et al. Exploration of large digital sky surveys. In *Mining the Sky*, pages 305–322. Springer, 2001.

[2] Kirk Borne. Virtual observatories, data mining, and astroinformatics. In *Planets, Stars and Stellar Systems*, pages 403–443. Springer, 2013.

[3] S George Djorgovski, Ashish Mahabal, Andrew Drake, Matthew Graham, and Ciro Donalek. Sky surveys. In *Planets, Stars and Stellar Systems*, pages 223–281. Springer, 2013.

[4] Nicholas Kaiser. Pan-starrs: a wide-field optical survey telescope array. In *Ground-based Telescopes*, volume 5489, pages 11–22. International Society for Optics and Photonics, 2004.

[5] HA Flewelling, EA Magnier, KC Chambers, JN Heasley, C Holmberg, ME Huber, W Sweeney, CZ Waters, T Chen, D Farrow, et al. The pan-starrs1 database and data products. *arXiv preprint arXiv:1612.05243*, 2016.

[6] Chien Y Peng, Luis C Ho, Chris D Impey, and Hans-Walter Rix. Detailed structural decomposition of galaxy images. *The Astronomical Journal*, 124(1):266, 2002.

[7] L Simard. Photometric redshifts and the luminosity-size relation of galaxies to z= 1. 1. In *Photometric Redshifts and the Detection of High Redshift Galaxies*, volume 191, page 325, 1999.

[8] Christopher J Conselice. The relationship between stellar light distributions of galaxies and their formation histories. *The Astrophysical Journal Supplement Series*, 147(1):1, 2003.

[9] Roberto G Abraham, Sidney Van Den Bergh, and Preethi Nair. A new approach to galaxy morphology. i. analysis of the sloan digital sky survey early data release. *The Astrophysical Journal*, 588(1):218, 2003.

[10] Lior Shamir. Ganalyzer: A tool for automatic galaxy image analysis. *The Astrophysical Journal*, 736(2):141, 2011.

[11] Darren R Davis and Wayne B Hayes. Sparcfire: Scalable automated detection of spiral galaxy arm segments. *The Astrophysical Journal*, 790(2):87, 2014.

[12] Preethi B Nair and Roberto G Abraham. A catalog of detailed visual morphological classifications for 14,034 galaxies in the sloan digital sky survey. *The Astrophysical Journal Supplement Series*, 186(2):427, 2010.

[13] Anthony Baillard, Emmanuel Bertin, Valérie De Lapparent, Pascal Fouqué, Stéphane Arnouts, Yannick Mellier, Roser Pelló, J-F Leborgne, Philippe Prugniel, Dmitry Makarov, et al. The efigi catalogue of 4458 nearby galaxies with detailed morphology. *Astronomy & Astrophysics*, 532:A74, 2011.

[14] Chris J Lintott, Kevin Schawinski, Anže Slosar, Kate Land, Steven Bamford, Daniel Thomas, M Jordan Raddick, Robert C Nichol, Alex Szalay, Dan Andreescu, et al. Galaxy zoo: morphologies derived from visual inspection of galaxies from the sloan digital sky survey. *Monthly Notices of the Royal Astronomical Society*, 389(3):1179–1189, 2008.

[15] Chris Lintott, Kevin Schawinski, Steven Bamford, Anže Slosar, Kate Land, Daniel Thomas, Edd Edmondson, Karen Masters, Robert C Nichol, M Jordan Raddick, et al. Galaxy zoo 1: data release of morphological classifications for nearly 900 000 galaxies. *Monthly Notices of the Royal Astronomical Society*, 410(1):166–178, 2011.

[16] Kyle W Willett, Chris J Lintott, Steven P Bamford, Karen L Masters, Brooke D Simmons, Kevin RV Casteels, Edward M Edmondson, Lucy F Fortson, Sugata Kaviraj, William C Keel, et al. Galaxy zoo 2: detailed morphological classifications for 304 122

galaxies from the sloan digital sky survey. *Monthly Notices of the Royal Astronomical Society*, page stt1458, 2013.

[17] Lior Shamir. Automatic morphological classification of galaxy images. *Monthly Notices of the Royal Astronomical Society*, 399(3):1367–1372, 2009.

[18] M Huertas-Company, L Tasca, D Rouan, D Pelat, J Kneib, O Le Fevre, P Capak, J Kartaltepe, A Koekemoer, H Mccracken, et al. A robust morphological classification of high-redshift galaxies using support vector machines on seeing limited images. *Astronomy and Astrophysics*, 497(3):743, 2009.

[19] Manda Banerji, Ofer Lahav, Chris J Lintott, Filipe B Abdalla, Kevin Schawinski, Steven P Bamford, Dan Andreescu, Phil Murray, M Jordan Raddick, Anze Slosar, et al. Galaxy zoo: reproducing galaxy morphologies via machine learning. *Monthly Notices of the Royal Astronomical Society*, 406(1):342–353, 2010.

[20] Lior Shamir, Anthony Holincheck, and John Wallin. Automatic quantitative morphological analysis of interacting galaxies. *Astronomy and Computing*, 2:67–73, 2013.

[21] Andrew Schutter and Lior Shamir. Galaxy morphology—an unsupervised machine learning approach. *Astronomy and Computing*, 12:60–66, 2015.

[22] Evan Kuminski, Joe George, John Wallin, and Lior Shamir. Combining human and machine learning for morphological analysis of galaxy images. *Publications of the Astronomical Society of the Pacific*, 126(944):959–967, 2014.

[23] Sander Dieleman, Kyle W Willett, and Joni Dambre. Rotation-invariant convolutional neural networks for galaxy morphology prediction. *Monthly Notices of the Royal Astronomical Society*, 450(2):1441–1459, 2015.

[24] Alex Hocking, James E Geach, Yi Sun, and Neil Davey. An automatic taxonomy of galaxy morphology using unsupervised machine learning. *Monthly Notices of the Royal Astronomical Society*, 473(1):1108–1129, 2017.

[25] Evan Kuminski and Lior Shamir. A hybrid approach to machine learning annotation of large galaxy image databases. *Astronomy and Computing*, 25:257–269, 2018.

[26] Pedro Silva, Leon Cao, and Wayne Hayes. Sparcfire: Enhancing spiral galaxy recognition using arm analysis and random forests. *Galaxies*, 6(3):95, 2018.

[27] Marc Huertas-Company, JA Aguerri, M Bernardi, S Mei, and J Sánchez Almeida. Revisiting the hubble sequence in the sdss dr7 spectroscopic sample: a publicly available bayesian automated classification. *arXiv preprint arXiv:1010.3018*, 2010.

[28] Luc Simard, J Trevor Mendel, David R Patton, Sara L Ellison, and Alan W McConnachie. A catalog of bulge+ disk decompositions and updated photometry for 1.12 million galaxies in the sloan digital sky survey. *The Astrophysical Journal Supplement Series*, 196(1):11, 2011.

[29] Lior Shamir and John Wallin. Automatic detection and quantitative assessment of peculiar galaxy pairs in sloan digital sky survey. *Monthly Notices of the Royal Astronomical Society*, 443(4):3528–3537, 2014.

[30] Evan Kuminski and Lior Shamir. Computer-generated visual morphology catalog of ∼3,000,000 sdss galaxies. *ApJS*, 223(2):20, 2016.

[31] M Huertas-Company, R Gravet, G Cabrera-Vives, PG Pérez-González, JS Kartaltepe, G Barro, M Bernardi, S Mei, F Shankar, P Dimauro, et al. A catalog of visual-like morphologies in the 5 candels fields using deep-learning. *arXiv preprint arXiv:1509.05429*, 2015.

[32] Marc Huertas-Company, Pablo G Pérez-González, Simona Mei, Francesco Shankar, Mariangela Bernardi, Emanuele Daddi, Guillermo Barro, Guillermo Cabrera-Vives, Andrea Cattaneo, Paola Dimauro, et al. The morphologies of massive galaxies from z˜ 3-witnessing the 2 channels of bulge growth. *arXiv preprint arXiv:1506.03084*, 2015.

[33] Ian Timmis and Lior Shamir. A catalog of automatically detected ring galaxy candidates in panstarss. *The Astrophysical Journal Supplement Series*, 231(1):2, 2017.

[34] Nicholas Paul, Nicholas Virag, and Lior Shamir. A catalog of photometric redshift and the distribution of broad galaxy morphologies. *Galaxies*, 6(2):64, 2018.

[35] Lior Shamir. Automatic detection of full ring galaxy candidates in sdss. *Monthly Notices of the Royal Astronomical Society*, 491(3):3767–3777, 2019.

[36] A Baldeschi, A Miller, M Stroh, R Margutti, and DL Coppejans. Star formation and morphological properties of galaxies in the pan-starrs 3π survey. i. a machine-learning approach to galaxy and supernova classification. *The Astrophysical Journal*, 902(1): 60, 2020.

[37] KW Hodapp, N Kaiser, H Aussel, W Burgett, KC Chambers, M Chun, T Dombeck, A Douglas, D Hafner, J Heasley, et al. Design of the pan-starrs telescopes. *AN*, 325 (6-8):636–642, 2004.

[38] Kenneth C Chambers, EA Magnier, N Metcalfe, HA Flewelling, ME Huber, CZ Waters, L Denneau, PW Draper, D Farrow, DP Finkbeiner, et al. The pan-starrs1 surveys. *arXiv preprint arXiv:1612.05560*, 2016.

[39] Nobuyuki Otsu. A threshold selection method from gray-level histograms. *IEEE Transactions on Systems, Man, and Cybernetics*, 9(1):62–66, 1979.

[40] Yann LeCun, Léon Bottou, Yoshua Bengio, Patrick Haffner, et al. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

[41] H Domínguez Sánchez, M Huertas-Company, M Bernardi, S Kaviraj, JL Fischer, TMC Abbott, FB Abdalla, J Annis, S Avila, D Brooks, et al. Transfer learning for galaxy morphology from one survey to another. *Monthly Notices of the Royal Astronomical Society*, 484(1):93–100, 2019.

[42] Kate Land, Anže Slosar, Chris Lintott, Dan Andreescu, Steven Bamford, Phil Murray, Robert Nichol, M Jordan Raddick, Kevin Schawinski, Alex Szalay, et al. Galaxy zoo:

the large-scale spin statistics of spiral galaxies in the sloan digital sky survey. *Monthly Notices of the Royal Astronomical Society*, 388(4):1686–1692, 2008.

[43] Levente Dojcsak and Lior Shamir. Quantitative analysis of spirality in elliptical galaxies. *New Astronomy*, 28:1–8, 2014.

[44] Lior Shamir, Nikita Orlov, D Mark Eckley, Tomasz Macura, Josiah Johnston, and Ilya G Goldberg. Wndchrm–an open source utility for biological image analysis. *Source Code for Biology and Medicine*, 3(1):1–13, 2008.

[45] Nolan Li and Ani R Thakar. Casjobs and mydb: A batch query workbench. *Computing in Science & Engineering*, 10(1):18–29, 2008.

[46] Asifullah Khan, Anabia Sohail, Umme Zahoora, and Aqsa Saeed Qureshi. A survey of the recent architectures of deep convolutional neural networks. *Artificial Intelligence Review*, 53(8):5455–5516, 2020.

[47] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 770–778, 2016.

[48] James A Nichols, Hsien W Herbert Chan, and Matthew AB Baker. Machine learning: applications of artificial intelligence to imaging and diagnosis. *Biophysical Reviews*, 11 (1):111–118, 2019.

[49] Tien T Tang, Janice A Zawaski, Kathleen N Francis, Amina A Qutub, and M Waleed Gaber. Image-based classification of tumor type and growth rate using machine learning: a preclinical study. *Scientific Reports*, 9(1):1–10, 2019.

[50] Carlos Martin-Isla, Victor M Campello, Cristian Izquierdo, Zahra Raisi-Estabragh, Bettina Baeßler, Steffen E Petersen, and Karim Lekadir. Image-based cardiac diagnosis with machine learning: a review. *Frontiers in Cardiovascular Medicine*, 7:1, 2020.

[51] Mahdieh Poostchi, Kamolrat Silamut, Richard J Maude, Stefan Jaeger, and George Thoma. Image analysis and machine learning for detecting malaria. *Translational Research*, 194:36–55, 2018.

[52] Nikita V Orlov, Ashani T Weeraratna, Stephen M Hewitt, Christopher E Coletta, John D Delaney, D Mark Eckley, Lior Shamir, and Ilya G Goldberg. Automatic detection of melanoma progression by histological analysis of secondary sites. *Cytometry Part A*, 81(5):364–373, 2012.

[53] Justin Ker, Yeqi Bai, Hwei Yee Lee, Jai Rao, and Lipo Wang. Automated brain histology classification using machine learning. *Journal of Clinical Neuroscience*, 66: 239–245, 2019.

[54] Arkadiusz Gertych, Nathan Ing, Zhaoxuan Ma, Thomas J Fuchs, Sadri Salman, Sambit Mohanty, Sanica Bhele, Adriana Velásquez-Vacca, Mahul B Amin, and Beatrice S Knudsen. Machine learning approaches to analyze histological images of tissues from radical prostatectomies. *Computerized Medical Imaging and Graphics*, 46:197–208, 2015.

[55] Ansi Zhang, Shaobo Li, Yuxin Cui, Wanli Yang, Rongzhi Dong, and Jianjun Hu. Limited data rolling bearing fault diagnosis with few-shot learning. *IEEE Access*, 7: 110895–110904, 2019.

[56] Jennifer Wortman Vaughan. Making better use of the crowd: How crowdsourcing can advance machine learning research. *J. Mach. Learn. Res.*, 18(1):7026–7071, 2017.

[57] Victor S Sheng and Jing Zhang. Machine learning with crowdsourcing: A brief summary of the past research and future directions. In *Proceedings of the AAAI conference on artificial intelligence*, volume 33, pages 9837–9843, 2019.

[58] Naihui Zhou, Zachary D Siegel, Scott Zarecor, Nigel Lee, Darwin A Campbell, Carson M Andorf, Dan Nettleton, Carolyn J Lawrence-Dill, Baskar Ganapathysubramanian, Jonathan W Kelly, et al. Crowdsourcing image analysis for plant phenomics to

generate ground truth data for machine learning. *PLoS computational biology*, 14(7): e1006337, 2018.

[59] Lior Shamir, Carol Yerby, Robert Simpson, Alexander M von Benda-Beckmann, Peter Tyack, Filipa Samarra, Patrick Miller, and John Wallin. Classification of large acoustic datasets using machine learning and crowdsourcing: Application to whale calls. *The Journal of the Acoustical Society of America*, 135(2):953–962, 2014.

[60] Matthew Lease. On quality control and machine learning in crowdsourcing. In *Workshops at the Twenty-Fifth AAAI Conference on Artificial Intelligence*, 2011.

[61] Li Fei-Fei, Rob Fergus, and Pietro Perona. One-shot learning of object categories. *IEEE transactions on pattern analysis and machine intelligence*, 28(4):594–611, 2006.

[62] Lior Wolf, Tal Hassner, and Yaniv Taigman. The one-shot similarity kernel. In *2009 IEEE 12th International Conference on Computer Vision*, pages 897–902. IEEE, 2009.

[63] Gregory Koch, Richard Zemel, Ruslan Salakhutdinov, et al. Siamese neural networks for one-shot image recognition. In *ICML deep learning workshop*, volume 2. Lille, 2015.

[64] Oriol Vinyals, Charles Blundell, Timothy Lillicrap, Daan Wierstra, et al. Matching networks for one shot learning. *Advances in Neural Information Processing Systems*, 29:3630–3638, 2016.

[65] Gabriel Bender, Pieter-Jan Kindermans, Barret Zoph, Vijay Vasudevan, and Quoc Le. Understanding and simplifying one-shot architecture search. In *International Conference on Machine Learning*, pages 550–559. PMLR, 2018.

[66] Xu Tang, Fanbo Meng, Xiangrong Zhang, Yiu-Ming Cheung, Jingjing Ma, Fang Liu, and Licheng Jiao. Hyperspectral image classification based on 3-d octave convolution with spatial–spectral attention network. *IEEE Transactions on Geoscience and Remote Sensing*, 59(3):2430–2447, 2020.

[67] Jing Bai, Bixiu Ding, Zhu Xiao, Licheng Jiao, Hongyang Chen, and Amelia C Regan. Hyperspectral image classification based on deep attention graph convolutional network. *IEEE Transactions on Geoscience and Remote Sensing*, 2021.

[68] Corinna Cortes and Vladimir Vapnik. Support-vector networks. *Machine learning*, 20 (3):273–297, 1995.

[69] Wun-Hwa Chen, Sheng-Hsun Hsu, and Hwang-Pin Shen. Application of svm and ann for intrusion detection. *Computers & Operations Research*, 32(10):2617–2634, 2005.

[70] Srinivas Mukkamala, Guadalupe Janoski, and Andrew Sung. Intrusion detection using neural networks and support vector machines. In *Proceedings of the 2002 International Joint Conference on Neural Networks. IJCNN'02 (Cat. No. 02CH37290)*, volume 2, pages 1702–1707. IEEE, 2002.

[71] Li Zhang, Weida Zhou, and Licheng Jiao. Wavelet support vector machine. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, 34(1):34–39, 2004.

[72] Kyung-Shik Shin, Taik Soo Lee, and Hyun-jung Kim. An application of support vector machines in bankruptcy prediction model. *Expert Systems with Applications*, 28(1): 127–135, 2005.

[73] Evgeny Byvatov, Uli Fechner, Jens Sadowski, and Gisbert Schneider. Comparison of support vector machine and artificial neural network systems for drug/nondrug classification. *Journal of chemical information and computer sciences*, 43(6):1882–1889, 2003.

[74] Licheng Jiao, Liefeng Bo, and Ling Wang. Fast sparse approximation for least squares support vector machine. *IEEE Transactions on Neural Networks*, 18(3):685–697, 2007.

[75] Joana S Paiva, João Cardoso, and Tânia Pereira. Supervised learning methods for pathological arterial pulse wave differentiation: a svm and neural networks approach. *International Journal of Medical Informatics*, 109:30–38, 2018.

[76] Vincent Dumoulin and Francesco Visin. A guide to convolution arithmetic for deep learning, 2018.

[77] Chih-Chung Chang and Chih-Jen Lin. Libsvm: a library for support vector machines. *ACM transactions on intelligent systems and technology (TIST)*, 2(3):1–27, 2011.

[78] Sameer A Nene, Shree K Nayar, Hiroshi Murase, et al. Columbia object image library (coil-100). Technical report, Columbia University, 1996.

[79] Jeremy Howard. Imagenette, 2020. URL https://github.com/fastai/imagenette/.

[80] Ezz El-Din Hemdan, Marwa A. Shouman, and Mohamed Esmail Karar. COVIDX-Net: A Framework of Deep Learning Classifiers to Diagnose COVID-19 in X-Ray Images. *arXiv e-prints*, art. arXiv:2003.11055, March 2020.

[81] Cedric Gangloff, Sonia Rafi, Guillaume Bouzillé, Louis Soulat, and Marc Cuggia. Machine learning is the key to diagnose covid-19: a proof-of-concept study. *Scientific Reports*, 11(1):7166, Mar 2021. ISSN 2045-2322. doi: 10.1038/s41598-021-86735-9. URL https://doi.org/10.1038/s41598-021-86735-9.

[82] Wei Tse Li, Jiayan Ma, Neil Shende, Grant Castaneda, Jaideep Chakladar, Joseph C. Tsai, Lauren Apostol, Christine O. Honda, Jingyue Xu, Lindsay M. Wong, Tianyi Zhang, Abby Lee, Aditi Gnanasekar, Thomas K. Honda, Selena Z. Kuo, Michael Andrew Yu, Eric Y. Chang, Mahadevan " Raj" Rajasekaran, and Weg M. Ongkeko. Using machine learning of clinical data to diagnose covid-19: a systematic review and meta-analysis. *BMC Medical Informatics and Decision Making*, 20(1):247, Sep 2020. ISSN 1472-6947. doi: 10.1186/s12911-020-01266-z. URL https://doi.org/10.1186/s12911-020-01266-z.

[83] Muhammad E. H. Chowdhury, Tawsifur Rahman, Amith Khandakar, Rashid Mazhar, Muhammad Abdul Kadir, Zaid Bin Mahbub, Khandakar Reajul Islam, Muhammad Salman Khan, Atif Iqbal, Nasser Al Emadi, Mamun Bin Ibne Reaz, and Mo-

hammad Tariqul Islam. Can ai help in screening viral and covid-19 pneumonia? *IEEE Access*, 8:132665–132676, 2020. doi: 10.1109/ACCESS.2020.3010287.

[84] Tawsifur Rahman, Amith Khandakar, Yazan Qiblawey, Anas Tahir, Serkan Kiranyaz, Saad Bin Abul Kashem, Mohammad Tariqul Islam, Somaya Al Maadeed, Susu M. Zughaier, Muhammad Salman Khan, and Muhammad E.H. Chowdhury. Exploring the effect of image enhancement techniques on covid-19 detection using chest x-ray images. *Computers in Biology and Medicine*, 132:104319, 2021. ISSN 0010-4825. doi: https://doi.org/10.1016/j.compbiomed.2021.104319. URL https://www.sciencedirect.com/science/article/pii/S001048252100113X.

[85] Hunter Goddard and Lior Shamir. A catalog of broad morphology of pan-starrs galaxies based on deep learning. *The Astrophysical Journal Supplement Series*, 251(2):28, 2020.

[86] Léon Bottou and Chih-Jen Lin. Support vector machine solvers. *Large scale kernel machines*, 3(1):301–320, 2007.

[87] MP Pavan Kumar, B Poornima, HS Nagendraswamy, C Manjunath, BE Rangaswamy, M Varsha, and HP Vinutha. Image abstraction framework as a pre-processing technique for accurate classification of archaeological monuments using machine learning approaches. *SN Computer Science*, 3(1):1–30, 2022.

[88] Fahad Shahbaz Khan, Shida Beigpour, Joost Van de Weijer, and Michael Felsberg. Painting-91: a large scale database for computational painting categorization. *Machine Vision and Applications*, 25(6):1385–1397, 2014.

[89] Kieran Jay Edwards and Mohamed Medhat Gaber. Astronomy and big data. *Studies in Big Data. Springer*, 2014.

[90] J Anthony Tyson. Cosmology data analysis challenges and opportunities in the lsst sky survey. In *Journal of Physics: Conference Series*, volume 1290, page 012001. IOP Publishing, 2019.

[91] Donald G York, J Adelman, John E Anderson Jr, Scott F Anderson, James Annis, Neta A Bahcall, JA Bakken, Robert Barkhouser, Steven Bastian, Eileen Berman, et al. The sloan digital sky survey: Technical summary. *Astronomical Journal*, 120(3):1579, 2000.

[92] Ivan K Baldry, Aaron SG Robotham, David T Hill, Simon P Driver, Jochen Liske, Peder Norberg, Steven P Bamford, Andrew M Hopkins, Jon Loveday, John A Peacock, et al. Galaxy and mass assembly (gama): the input catalogue and star–galaxy separation. *Monthly Notices of the Royal Astronomical Society*, 404(1):86–100, 2010.

[93] EC Vasconcellos, RR De Carvalho, RR Gal, FL LaBarbera, HV Capelato, H Frago Campos Velho, Marina Trevisan, and RSR Ruiz. Decision tree classifiers for star/galaxy separation. *Astronomical Journal*, 141(6):189, 2011.

[94] András Kovács and István Szapudi. Star–galaxy separation strategies for wise-2mass all-sky infrared galaxy catalogues. *Monthly Notices of the Royal Astronomical Society*, 448(2):1305–1313, 2015.

[95] PO Baqui, V Marra, L Casarini, R Angulo, LA Díaz-García, C Hernández-Monteagudo, PAA Lopes, C López-Sanjuan, D Muniesa, VM Placco, et al. The minijpas survey: star-galaxy classification using machine learning. *Astronomy & Astrophysics*, 645:A87, 2021.

[96] Gabriel B Brammer, Pieter G van Dokkum, and Paolo Coppi. Eazy: a fast, public photometric redshift code. *Astrophysical Journal*, 686(2):1503, 2008.

[97] Mara Salvato, Olivier Ilbert, and Ben Hoyle. The many flavours of photometric redshifts. *Nature Astronomy*, 3(3):212–222, 2019.

[98] Angus H Wright, Hendrik Hildebrandt, Jan Luca Van den Busch, and Catherine Heymans. Photometric redshift calibration with self-organising maps. *Astronomy & Astrophysics*, 637:A100, 2020.

[99] S Fotopoulou and S Paltani. Cpz: Classification-aided photometric-redshift estimation. *Astronomy & Astrophysics*, 619:A14, 2018.

[100] D Wittman. What lies beneath: Using p (z) to reduce systematic photometric redshift errors. *ApJ*, 700(2):L174, 2009.

[101] Gary Bernstein and Dragan Huterer. Catastrophic photometric redshift errors: weak-lensing survey requirements. *MNRAS*, 401(2):1399–1408, 2010.

[102] Tomas Dahlen, Bahram Mobasher, Sandra M Faber, Henry C Ferguson, Guillermo Barro, Steven L Finkelstein, Kristian Finlator, Adriano Fontana, Ruth Gruetzbauch, Seth Johnson, et al. A critical assessment of photometric redshift methods: a candels investigation. *ApJ*, 775(2):93, 2013.

[103] Markus Michael Rau, Stella Seitz, Fabrice Brimioulle, Eibe Frank, Oliver Friedrich, Daniel Gruen, and Ben Hoyle. Accurate photometric redshift probability density estimation–method comparison and application. *MNRAS*, 452(4):3710–3725, 2015.

[104] Masayuki Tanaka, Jean Coupon, Bau-Ching Hsieh, Sogo Mineo, Atsushi J Nishizawa, Joshua Speagle, Hisanori Furusawa, Satoshi Miyazaki, and Hitoshi Murayama. Photometric redshifts for hyper suprime-cam subaru strategic program data release 1. *PASJ*, 70(SP1):S9, 2018.

[105] Ting-Yun Cheng, Christopher J Conselice, Alfonso Aragón-Salamanca, Nan Li, Asa FL Bluck, Will G Hartley, James Annis, David Brooks, Peter Doel, Juan García-Bellido, et al. Optimizing automatic morphological classification of galaxies with machine learning and deep learning using dark energy survey imaging. *Monthly Notices of the Royal Astronomical Society*, 493(3):4209–4228, 2020.

[106] H Domínguez Sánchez, M Huertas-Company, M Bernardi, D Tuccillo, and JL Fischer. Improving galaxy morphologies for sdss with deep learning. *Monthly Notices of the Royal Astronomical Society*, 476(3):3661–3676, 2018.

[107] Asad Khan, EA Huerta, Sibo Wang, Robert Gruendl, Elise Jennings, and Huihuo Zheng. Deep learning at scale for the construction of galaxy catalogs in the dark energy survey. *Physics Letters B*, 795:248–258, 2019.

[108] Lior Shamir. Automatic morphological classification of galaxy images. *Monthly Notices of the Royal Astronomical Society*, 399(3):1367–1372, 2009.

[109] Roberto E González, Roberto P Munoz, and Cristian A Hernández. Galaxy detection and identification using deep learning and data augmentation. *Astronomy and Computing*, 25:103–109, 2018.

[110] PH Barchi, RR de Carvalho, RR Rosa, RA Sautter, M Soares-Santos, BAD Marques, E Clua, TS Gonçalves, C de Sá-Freitas, and TC Moura. Machine and deep learning applied to galaxy morphology-a comparative study. *Astronomy and Computing*, 30: 100334, 2020.

[111] Manda Banerji, Ofer Lahav, Chris J Lintott, Filipe B Abdalla, Kevin Schawinski, Steven P Bamford, Dan Andreescu, Phil Murray, M Jordan Raddick, Anze Slosar, et al. Galaxy zoo: reproducing galaxy morphologies via machine learning. *Monthly Notices of the Royal Astronomical Society*, 406(1):342–353, 2010.

[112] Sebastian Lapuschkin, Stephan Wäldchen, Alexander Binder, Grégoire Montavon, Wojciech Samek, and Klaus-Robert Müller. Unmasking clever hans predictors and assessing what machines really learn. *Nature communications*, 10(1):1–8, 2019.

[113] Patrick Schramowski, Wolfgang Stammer, Stefano Teso, Anna Brugger, Franziska Herbert, Xiaoting Shao, Hans-Georg Luigs, Anne-Katrin Mahlein, and Kristian Kersting. Making deep neural networks right for the right scientific reasons by interacting with their explanations. *Nature Machine Intelligence*, 2(8):476–486, 2020.

[114] Rehab Ali Ibrahim, Mohamed Abd Elaziz, Ahmed A Ewees, Ibrahim M Selim, and Songfeng Lu. Galaxy images classification using hybrid brain storm optimization with

moth flame optimization. *Journal of Astronomical Telescopes, Instruments, and Systems*, 4(3):038001, 2018.

[115] Antonio Torralba and Alexei A Efros. Unbiased look at dataset bias. In *CVPR 2011*, pages 1521–1528. IEEE, 2011.

[116] Sanchari Dhar and Lior Shamir. Evaluation of the benchmark datasets for testing the efficacy of deep convolutional neural networks. *Visual Informatics*, 5(3):92–101, 2021.

[117] Lior Shamir. Evaluation of face datasets as tools for assessing the performance of face recognition methods. *International Journal of Computer Vision*, 79(3):225–230, 2008.

[118] Sanchari Dhar and Lior Shamir. Systematic biases when using deep neural networks for annotating large catalogs of astronomical images. *A&C*, 38:100545, 2022.

[119] François Lanusse et al. The dawes review 10: The impact of deep learning for the analysis of galaxy surveys. *Publications of the Astronomical Society of Australia*, 40: e01, 2023.

[120] Jan Kremer, Kristoffer Stensbo-Smidt, Fabian Gieseke, Kim Steenstrup Pedersen, and Christian Igel. Big universe, big data: machine learning and image analysis for astronomy. *IEEE Intelligent Systems*, 32(2):16–22, 2017.

[121] Galen Harrison, Julia Hanson, Christine Jacinto, Julio Ramirez, and Blase Ur. An empirical study on the perceived fairness of realistic, imperfect machine learning models. In *Proceedings of the 2020 conference on fairness, accountability, and transparency*, pages 392–402, 2020.

[122] Evan Kuminski and Lior Shamir. A computer-generated visual morphology catalog of 3,000,000 sdss galaxies. *The Astrophysical Journal Supplement Series*, 223(2):20, 2016.

[123] Brandon Carter, Siddhartha Jain, Jonas Mueller, and David Gifford. Overinterpretation reveals image classification model pathologies. *arXiv:2003.08907*, 2020.

[124] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: a simple way to prevent neural networks from overfitting. *The journal of machine learning research*, 15(1):1929–1958, 2014.

[125] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization, 2014. URL https://arxiv.org/abs/1412.6980.

[126] Larry McNish. The rasc calgary centre - the constellations, Oct 2005. URL https://calgary.rasc.ca/constellation.htm#list.

[127] Lior Shamir, Shari M Ling, William Scott, Marc Hochberg, Luigi Ferrucci, and Ilya G Goldberg. Early detection of radiographic knee osteoarthritis using computer-aided analysis. *Osteoarthritis and Cartilage*, 17(10):1307–1312, 2009.

[128] Alister W Graham. A galaxy classification grid that better recognizes early-type galaxy morphology. *Monthly Notices of the Royal Astronomical Society*, 487(4):4995–5009, 2019.

[129] Ansh Mittal, Anu Soorya, Preeti Nagrath, and D Jude Hemanth. Data augmentation based morphological classification of galaxies using deep convolutional neural network. *Earth Science Informatics*, pages 1–17, 2019.

[130] KM Hosny, MA Elaziz, IM Selim, and MM Darwish. Classification of galaxy color images using quaternion polar complex exponential transform and binary stochastic fractal search. *Astronomy and Computing*, page 100383, 2020.

[131] Hubert Cecotti. Rotation invariant descriptors for galaxy morphological classification. *International Journal of Machine Learning and Cybernetics*, pages 1–15, 2020.

[132] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

[133] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

[134] Zhilin Yang, Zihang Dai, Yiming Yang, Jaime Carbonell, Russ R Salakhutdinov, and Quoc V Le. Xlnet: Generalized autoregressive pretraining for language understanding. *Advances in neural information processing systems*, 32, 2019.

[135] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901, 2020.

[136] Michael Glass, Alfio Gliozzo, Rishav Chakravarti, Anthony Ferritto, Lin Pan, GP Shrivatsa Bhargav, Dinesh Garg, and Avirup Sil. Span selection pre-training for question answering. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 2773–2782, 2020.

[137] Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. Exploring the limits of transfer learning with a unified text-to-text transformer. *The Journal of Machine Learning Research*, 21(1):5485–5551, 2020.

[138] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020.

[139] Hugo Touvron, Matthieu Cord, Matthijs Douze, Francisco Massa, Alexandre Sablayrolles, and Hervé Jégou. Training data-efficient image transformers & distillation through attention. In *International conference on machine learning*, pages 10347–10357. PMLR, 2021.

[140] Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jégou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. Emerging properties in self-supervised vision transformers. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 9650–9660, 2021.

[141] Mahmoud Assran, Mathilde Caron, Ishan Misra, Piotr Bojanowski, Armand Joulin, Nicolas Ballas, and Michael Rabbat. Semi-supervised learning of visual features by non-parametrically predicting view assignments with support samples. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 8443–8452, 2021.

[142] Mike Walmsley, Chris Lintott, Tobias Géron, Sandor Kruk, Coleman Krawczyk, Kyle W Willett, Steven Bamford, Lee S Kelvin, Lucy Fortson, Yarin Gal, et al. Galaxy zoo decals: Detailed visual morphology measurements from volunteers and deep learning for 314 000 galaxies. *Monthly Notices of the Royal Astronomical Society*, 509(3): 3966–3988, 2022.

[143] Nour Eldeen Khalifa, Mohamed Hamed Taha, Aboul Ella Hassanien, and Ibrahim Selim. Deep galaxy v2: Robust deep convolutional neural networks for galaxy morphology classifications. In *2018 International Conference on Computing Sciences and Engineering (ICCSE)*, pages 1–6. IEEE, 2018.

[144] Kyle W Willett, Chris J Lintott, Steven P Bamford, Karen L Masters, Brooke D Simmons, Kevin RV Casteels, Edward M Edmondson, Lucy F Fortson, Sugata Kaviraj, William C Keel, et al. Galaxy zoo 2: detailed morphological classifications for 304 122 galaxies from the sloan digital sky survey. *Monthly Notices of the Royal Astronomical Society*, 435(4):2835–2860, 2013.

[145] Ross E Hart, Steven P Bamford, Kyle W Willett, Karen L Masters, Carolin Cardamone, Chris J Lintott, Robert J Mackay, Robert C Nichol, Christopher K Rosslowe, Brooke D Simmons, et al. Galaxy zoo: comparing the demographics of spiral arm

number and a new method for correcting redshift bias. *Monthly Notices of the Royal Astronomical Society*, 461(4):3663–3682, 2016.

[146] Kyle W Willett, Melanie A Galloway, Steven P Bamford, Chris J Lintott, Karen L Masters, Claudia Scarlata, Brooke D Simmons, Melanie Beck, Carolin N Cardamone, Edmond Cheung, et al. Galaxy zoo: morphological classifications for 120 000 galaxies in hst legacy imaging. *Monthly Notices of the Royal Astronomical Society*, 464(4): 4176–4203, 2016.

[147] Brooke D Simmons, Chris Lintott, Kyle W Willett, Karen L Masters, Jeyhan S Kartaltepe, Boris Häußler, Sugata Kaviraj, Coleman Krawczyk, SJ Kruk, Daniel H McIntosh, et al. Galaxy zoo: quantitative visual morphological classifications for 48,000 galaxies from candels. *Monthly Notices of the Royal Astronomical Society*, page stw2587, 2016.

[148] Mike Walmsley, Anna MM Scaife, Chris Lintott, Michelle Lochner, Verlon Etsebeth, Tobias Géron, Hugh Dickinson, Lucy Fortson, Sandor Kruk, Karen L Masters, et al. Practical galaxy morphology tools from deep supervised representation learning. *Monthly Notices of the Royal Astronomical Society*, 513(2):1581–1599, 2022.

[149] Mingxing Tan and Quoc Le. Efficientnet: Rethinking model scaling for convolutional neural networks. In *International conference on machine learning*, pages 6105–6114. PMLR, 2019.

[150] Vikas Jain, Nirbhay Modhe, and Piyush Rai. Scalable generative models for multi-label learning with missing labels. In *International Conference on Machine Learning*, pages 1636–1644. PMLR, 2017.

[151] Sanjay Kumar and Reshma Rastogi. Low rank label subspace transformation for multi-label learning with missing labels. *Information Sciences*, 596:53–72, 2022.

[152] Jun Huang, Linchuan Xu, Kun Qian, Jing Wang, and Kenji Yamanishi. Multi-label learning with missing and completely unobserved labels. *Data Mining and Knowledge Discovery*, 35:1061–1086, 2021.

[153] Elijah Cole, Oisin Mac Aodha, Titouan Lorieul, Pietro Perona, Dan Morris, and Nebojsa Jojic. Multi-label learning from single positive labels. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 933–942, 2021.

[154] Xiao Shi Huang, Felipe Perez, Jimmy Ba, and Maksims Volkovs. Improving transformer optimization through better initialization. In *International Conference on Machine Learning*, pages 4475–4483. PMLR, 2020.

[155] Noam Shazeer, Azalia Mirhoseini, Krzysztof Maziarz, Andy Davis, Quoc Le, Geoffrey Hinton, and Jeff Dean. Outrageously large neural networks: The sparsely-gated mixture-of-experts layer. *arXiv preprint arXiv:1701.06538*, 2017.

[156] Carlos Riquelme, Joan Puigcerver, Basil Mustafa, Maxim Neumann, Rodolphe Jenatton, André Susano Pinto, Daniel Keysers, and Neil Houlsby. Scaling vision with sparse mixture of experts. *Advances in Neural Information Processing Systems*, 34:8583–8595, 2021.

[157] Yanqi Zhou, Tao Lei, Hanxiao Liu, Nan Du, Yanping Huang, Vincent Zhao, Andrew M Dai, Quoc V Le, James Laudon, et al. Mixture-of-experts with expert choice routing. *Advances in Neural Information Processing Systems*, 35:7103–7114, 2022.

[158] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*, 2017.

[159] Sashank J Reddi, Satyen Kale, and Sanjiv Kumar. On the convergence of adam and beyond. *arXiv preprint arXiv:1904.09237*, 2019.

[160] Khaled Bayoudh, Raja Knani, Fayçal Hamdaoui, and Abdellatif Mtibaa. A survey on deep multimodal learning for computer vision: advances, trends, applications, and datasets. *The Visual Computer*, pages 1–32, 2021.