Deep Learning Project

MASTER DEGREE PROGRAM IN DATA SCIENCE AND ADVANCED ANALYTICS

Image Classification for Dermatology Problems

Group 19

Diogo Pires, 20230534
Majd Al Ajlani, 20230767
Manuel Gonçalves, 20230466
Sebastião Oliveira, 20220558
Stepan Kuznetsov, 20231002

April, 2024

NOVA Information Management School
Instituto Superior de Estatística e Gestão de Informação
Universidade Nova de Lisboa

INDEX

1. Introduction

Since Alex Krizhevsky's victory in the *ImageNet Computer Vision 2012* competition by using a Convolutional Neural Network (CNN), the development of new methods within this approach has undergone a rapid evolution, offering now diverse practical applications across numerous areas[1].

One particular field that has benefited from this evolution is Medicine. Nowadays, there are multiple Deep Learning model applications like Medical Imaging Diagnosis, Pathology & Histology, and Electronic Health Records (EHR) Analysis, among many others. These applications potentiate the improvement of patient care, can enhance the accuracy of diagnostics, and improve the overall efficiency of the healthcare system[2].

In this report we'll be focusing on Medical Imaging, in particular Image Classification for Dermatology Problems. Using images from the "*FITZPATRICK17"* repository, we'll walk through the process of building and testing CNNs to enhance diagnostic accuracy, and cross-checking along with more balanced metrics like the Weighted F1-score, to minimize the classification error when presented with multiple different skin conditions. For this task we mainly used *Tensorflow* and *Keras* libraries.

2. Data Exploration and Preprocessing

2.1. Exploratory Data Analysis

The "*FITZPATRICK17"* dataset has a total of 16577 values where, given the individual URLs, we'll be able to extract our images depicting different dermatology problems. The features *"mdh5ash"* which served an identification purpose not relevant to our analysis, and *"qc"* which was composed of around 97% missing values, were dropped.

There are two attributes dedicated to classifying skin pigmentation, *"fitzpatrick_scale"* and *"fitzpatrick_centaur"* annotated by the companies Scale AI and Centaur Labs respectively[3]. For our analysis, we chose to use the *"fitzpatrick_scale"*, because its values were more evenly distributed, almost resembling a normal distribution, though still with a higher representation of lighter skin tones.

There are three different label features: *"three_partition_label"*; *"nine_partition_label"*; and *"label"*. For this project, we'll use *"label"* as our target variable, as we find it to be the most relevant for classification. There are a total of 114 classes, with some classes having as much as 653 images whilst others have 53, this imbalance means that classification for conditions with less available data will prove to be more difficult, nevertheless we later explore strategies to mitigate this.

2.2. Image Extraction Process

We used the URLs present in the dataset to download the images to a local folder. For the rows with missing URLs, we used the alphanumeric URL with Python requests and some hard coding to extract and download all the available images. Aside from 10 broken URLs, we were able to extract a total of 16567 images into the local folder. In order to link the images stored in this folder to our dataset, we created a new column *"image_path"*.

2.3. Stratified Data Split & Feature Engineering

Given the noted imbalances in both the *"fitzpatrick_scale"* and *"label",* we decided that when performing the Train-Test split we would stratify by both these categories in order for every target condition to be proportionally represented in the Training, Validation and Test sets.

First, we created a new feature called *"fitzpatrick_range"* which was a grouping of *"fitzpatrick_scale"* values. The goal was to correct for values outside the scale, which we deemed as unknown, and to group lighter and darker skin tones in a way we would have a more proportional representation of both, also given their respective "*label".* By doing this, we wanted the model to have more conjoined data, for each skin condition, especially for the underrepresented darker skin tones to increase classification accuracy.

We also explored the option of creating different datasets using iterative stratification, a solution given the low number of some categories in the stratification target variable, with a new feature combination of "*fitzpatrick_scale"* and "*label"* and then extract the images from each data frame individually. However, we decided to perform the split by stratifying by "*fitzpatrick_range"* and "*label"* as separate features, 70% for Training, 15% for Validation and 15% for Test. We opted for this because it gave us the maneuvering space to test out different possible approaches on the raw image data, without being conditioned by a split from the beginning. Using the *"image_path"* we split the images of our local folder into Training, Validation and Test folders following the same stratification, where each image is stored inside the corresponding "*label"*.

3. Classification Model

3.1. Different CNNs Architectures Tested

Throughout the development of this project, we went back and forth on preprocessing and evaluation, testing multiple models with different architectures. We wanted our model to be useful without having the need to possess medical expertise to operate it, therefore we didn't opt for a multi-input model using other attributes like "*three_partition_label"* and "*nine_partition_label"*. The different models tested only take as input the data extracted from the images themselves.

Starting with our Baseline CNN model, using an input shape of 256 x 256 x 3, we begin the data augmentation process with five layers that include rescaling, random flipping, contrast adjustment, translation, and rotation of the images. The remaining model architecture is composed by four pairs of convolutional (with 3x3 filters and *ReLU* activation function) and max-pooling layers (with 2x2 filters). We gradually increase the units of each convolutional layer (32-64-128-256). Following this, we add a global average pooling and a flatten layer, followed by two pairs of dense layers, with 256 units and *ReLU* activation each, and dropout layers with a 25% chance of dropping. Finally, we finish with a dense layer with 114 units and a *softmax* function. The model was trained using the Adam optimizer with a 0.001 learning rate, measured the loss with categorical cross-entropy and evaluated using accuracy and weighted F1-score.

Using this model as a baseline for our next trials, we tested new approaches with transfer learning using different pre-trained models, such as: *VGG-16*; *VGG-19*; *ResNet50V2* and *InceptionV3*. We chose

these models because of the extensive existing literature on their successful applications on Medical Imaging Classifications[4].

Within the field of image classification, transfer learning is associated with using a model with a predefined structure that has been previously trained on a large dataset. Later on, it's fine-tuned by modifying the last few layers or retraining them completely given the specifics of the new classification task[5]. For our case, we used the same initial data augmentation layers as input for the transfer learning models, then for the final layers, a similar structure to the last 5 layers of our Baseline model. We found this to be a good fit for achieving solid results given the dataset's complexities. All transfer learning models we tested were pre-trained on the *ImageNet* dataset.

Let's start by breaking down the *ResNet50V2*. Residual Network (*ResNet*), according to its original paper, aims to "ease the training of networks that are substantially deep" by forming networks of stacked residual blocks [6,7]. The *V2* version has 50 layers and, relative to *V1*, underwent alteration in the way the links between blocks were formulated for propagation in order to improve training effectiveness[8].

The *VGG 16* & *19*, named after the Visual Geometry Group from Oxford, aimed at evaluating networks of increasing depth using an architecture with very small (3x3) convolution filters. They found that increasing the depth to 16 and 19 weight layers (hence the numbers in their names) can result in a notable model performance that generalizes well to other datasets[9].

InceptionV3 was developed by a team of researchers at Google. Based on the paper *"Rethinking the Inception Architecture for Computer Vision"* [10], they wanted to develop a model that would be more efficient and reach higher levels of accuracy on the *ImageNet* dataset. The model itself is made up of symmetric and asymmetric building blocks with an extensive use of batch normalization also applied to activation inputs[11].

3.2. Metrics Evaluation

We chose the weighted F1-score and accuracy as metrics for the evaluation of our models. The weighted F1-Score is a good metric for this project because it takes class imbalance into account, considering the proportion for each label in the dataset. We also kept the accuracy metric as comparison to other studies done using the "*FITZPATRICK17"* dataset.

In Appendix A, the bar chart elucidates the weighted F1-scores of our deep learning models. We can see that *ResNet50V2* outperforms the rest, with the other models showing varied results, with *VGG-19* trailing behind, reflecting a need for further optimization.

Appendix B contains a bar chart detailing the accuracy of the evaluated deep learning models. *ResNet50V2* also demonstrates the highest accuracy, affirming its robustness on our dataset. The chart also reveals a progressive decline in accuracy from *InceptionV3* to *VGG-19*, suggesting that certain architectures may not be as effective in this context.

3.3. Final Model

Our best model for image classification on this dataset was a transfer learning model based on *ResNet50V2* architecture. To enhance model robustness, we incorporated a series of data augmentation techniques, as mentioned.

The architecture integrates a Global Average Pooling layer followed by multiple dense layers. These are mixed with dropout layers to prevent overfitting, thus improving the model's ability to generalize effectively to new, unseen data. The final layer uses a *softmax* activation function, to output a probability distribution across the 114 classes, facilitating precise multi-class image classification.

Before fine-tuning, this model was compiled using the Adam optimizer, chosen for its efficiency in handling sparse gradients and adapting learning rates, and with the *categorical_crossentropy* loss function, which is standard for multi-class classification tasks. This configuration lets us achieve higher accuracy and optimize model performance.

The *ResNet50V2* model demonstrated our best performance with a test accuracy of approximately 0.3182, complemented by a test F1-score of roughly 0.3028, though the test loss stands at a high value of 2.9766.

To optimize this model for image classification, a *RandomSearch* tuner was utilized to navigate the hyperparameter space. The goal was to enhance the model's performance in terms of its weighted F1-score on a validation dataset. The search space for hyperparameters includes the choice of optimizer (Adam, RMSprop, or SGD) and a learning rate that varies logarithmically between 0.001 and 0.1.

A custom *"create_model_hypertune"* function assembles the *ResNet50v2* architecture with the chosen optimizer and metrics. This model is then trained and validated using a specified training data generator and validation data generator over a possible 200 epochs. The training process is regulated by callbacks including an early stopping mechanism and a checkpoint system, which help to prevent overfitting and ensure the best model state is preserved. After the conclusion of 10 trials, the hyperparameter search has identified an optimizer configuration that yields a maximum validation F1-score of 0.3285 with SGD as the optimizer with a 0.0153 as learning rate.

3.4. Error Analysis

We created a heat map-based confusion matrix on all conditions for our best model predictions, where we have the predicted label on the x-axis and the true label on the y-axis as seen in Appendix C. It's clear to see, given the evidence of darker colors on the diagonal, that there's a prevalence of correctly made predictions where the true label is equal to the predicted label.

However, to cross-check the results we ran the same confusion matrix on the 5 conditions with the most data, as seen in Appendix D. The most common misclassification error is associated with condition number 86. The model makes a lot of predictions for this condition and a lot of them are accurate because there are a lot of examples for it, yet due to the existing data imbalance, the model also wrongly predicts 86 for an extensive number of different conditions. By looking at the examples in Appendix E, we can likely conclude that these most common misclassifications were due to imbalanced data since the model isn't trained with a proportional amount of data for each condition.

4. Conclusion

The usage of deep learning and image classification in medicine and healthcare has become significantly more prominent in recent years, in order to improve patient care and enhance the accuracy of diagnostics. For this project, using the images from the "*FITZPATRICK17"* repository, multiple deep convolutional neural networks models and architectures were tested for multi-class classification. After considering the imbalanced representation, we made sure to split the data according to both target labels and Fitzpatrick scale to ensure diversity and at least one example of each category in the training, validation, and test datasets. Also, using data augmentation techniques allowed us to artificially increase the training set.

By training an image classifier using a transfer learning model based on *ResNet50V2* architecture, pre-trained on *ImageNet*, we achieved accuracy and weighted F1-score results of 0.32 and 0.30, respectively. After fine-tuning, using *Keras Tuner*, we were able to increase the weighted F1-score to 0.33.

5. Future Work

For this project we were given a vast dataset with good quality images, but we believe that updating it to include a more diverse representation of skin tones, particularly individuals with Fitzpatrick scale ratings above 3, could boost the algorithm's performance. One of the challenges of this project was also the limitations in hardware, so if we had more time and better resources (e.g. better GPUs or access to powerful virtual machines) it would be possible to test deeper and more complex architectures to get better results. Additionally, expanding the dataset with more data (e.g. patient's age, gender, and race), and applying a multiple input approach using the numerical and categorical features associated with each image, could be an interesting approach for future studies.

6. References

1. Deshpande A. A Beginner's Guide To Understanding Convolutional Neural Networks. UCLA. September 6, 2016. Accessed April 12, 2024. https://www.kdnuggets.com/2016/09/beginners-guide-understanding-convolutional-neural-networks-part-1.html.

2. Yang S, Zhu F, Ling X, Liu Q, Zhao P. Intelligent Health Care: Applications of Deep Learning in Computational Medicine. *Front Genet*. 2021;12. doi:10.3389/fgene.2021.607471

3. Groh M, Harris C, Soenksen L, et al. Evaluating Deep Neural Networks Trained on Clinical Images in Dermatology with the Fitzpatrick 17k Dataset. Published online April 20, 2021. http://arxiv.org/abs/2104.09957

4. Kim HE, Cosa-Linan A, Santhanam N, Jannesari M, Maros ME, Ganslandt T. Transfer learning for medical image classification: a literature review. *BMC Med Imaging*. 2022;22(1). doi:10.1186/s12880-022-00793-7

5. Buhl N. Training vs. Fine-tuning: What is the Difference? Encord. November 7, 2023. Accessed April 25, 2024. https://encord.com/blog/training-vs-fine-tuning/.

6. Datagen. ResNet-50: The Basics and a Quick Tutorial. Accessed April 20, 2024. https://datagen.tech/guides/computer-vision/resnet-50/#.

7. He K, Zhang X, Ren S, Sun J. Deep Residual Learning for Image Recognition. Published online December 10, 2015. http://arxiv.org/abs/1512.03385

8. Rahimzadeh M, Attar A. A modified deep convolutional neural network for detecting COVID-19 and pneumonia from chest X-ray images based on the concatenation of Xception and ResNet50V2. *Inform Med Unlocked*. 2020;19. doi:10.1016/j.imu.2020.100360

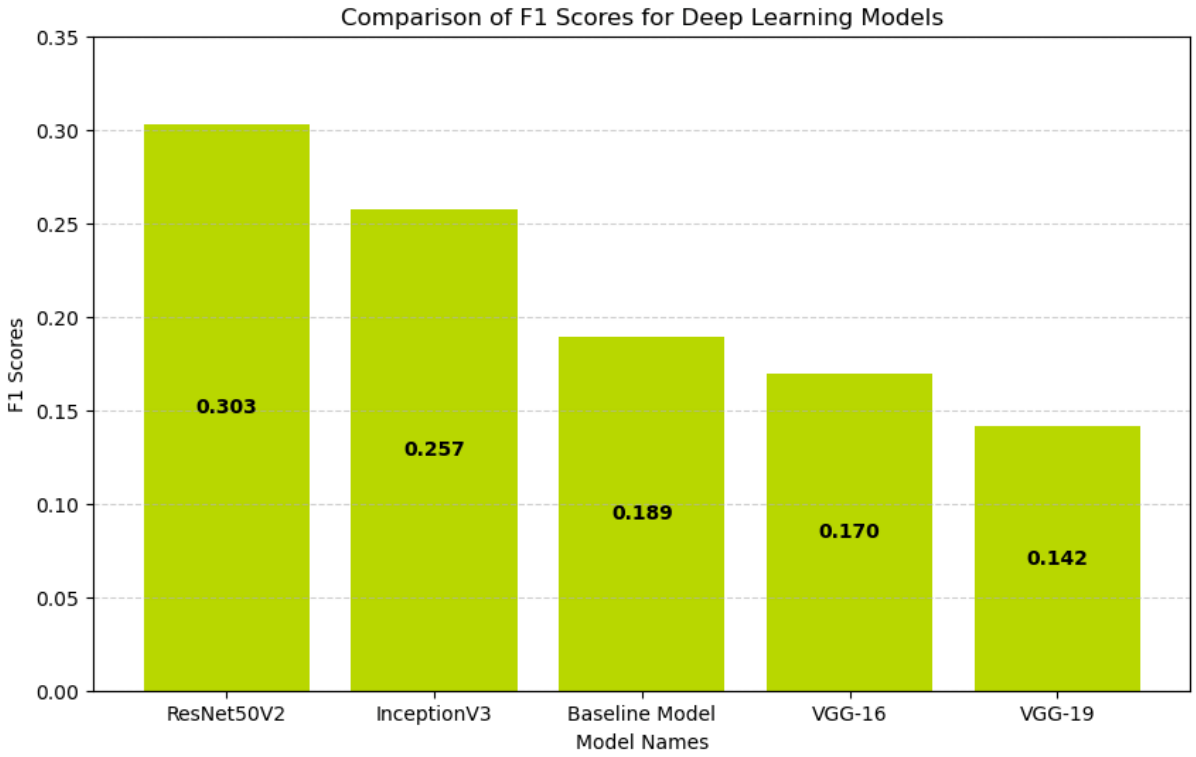
9. Simonyan K, Zisserman A. Very Deep Convolutional Networks for Large-Scale Image Recognition. Published online September 4, 2014. http://arxiv.org/abs/1409.1556

10. Szegedy C, Vanhoucke V, Ioffe S, Shlens J, Wojna Z. Rethinking the Inception Architecture for Computer Vision. Published online December 1, 2015. http://arxiv.org/abs/1512.00567

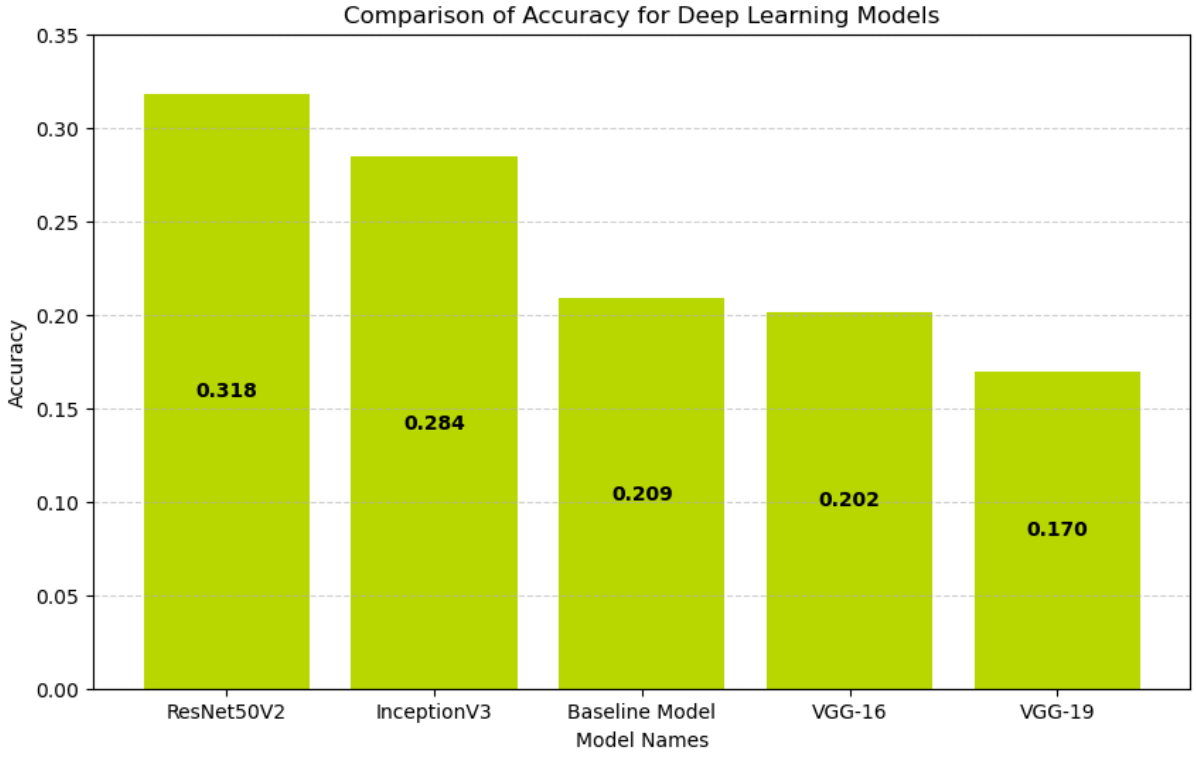
11. Cloud TPU. Advanced Guide to Inception v3. Accessed April 25, 2024. https://cloud.google.com/tpu/docs/inception-v3-advanced?hl.

Appendix

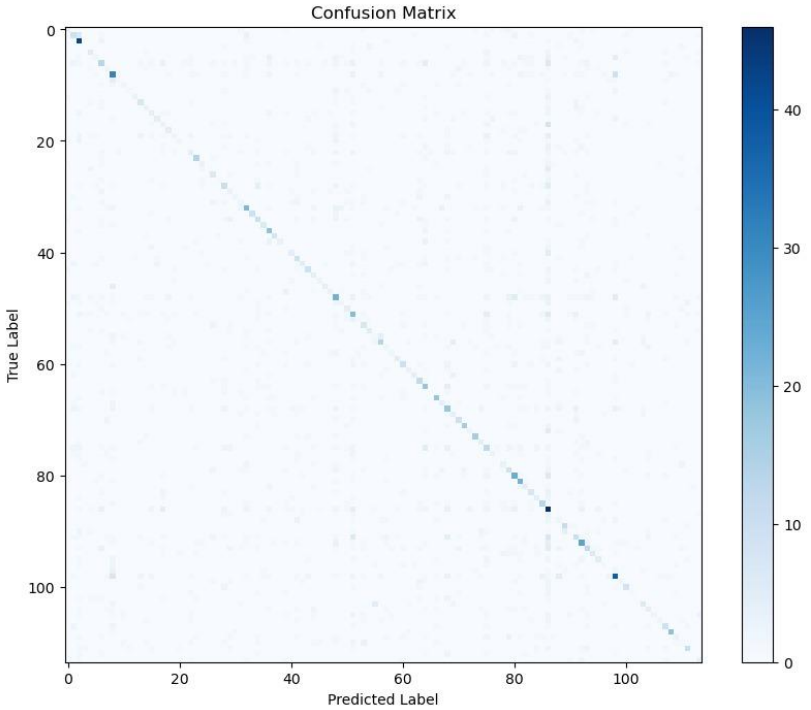
Appendix A



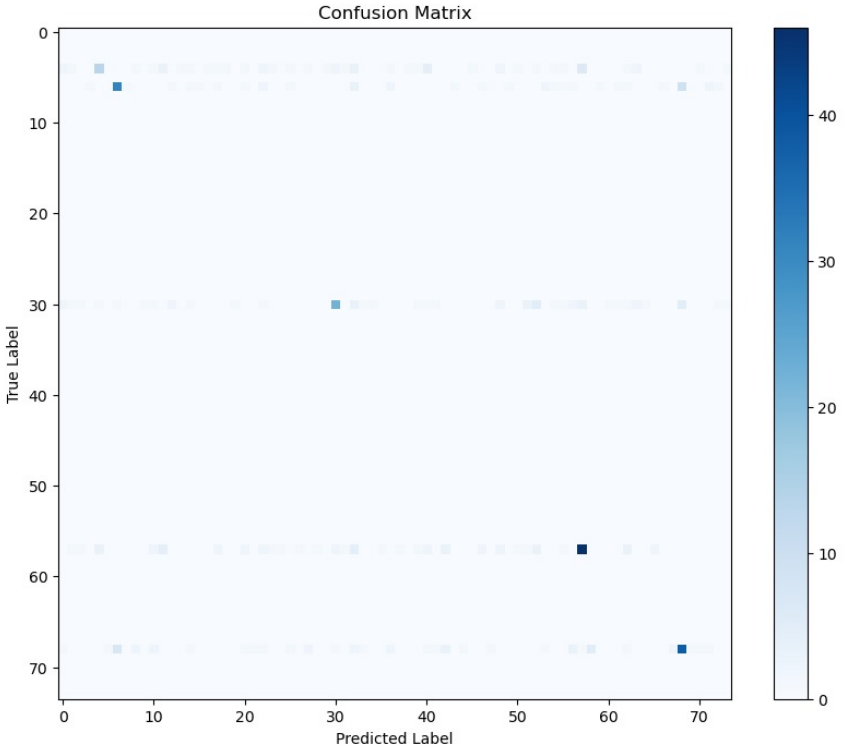
Appendix B



Appendix C

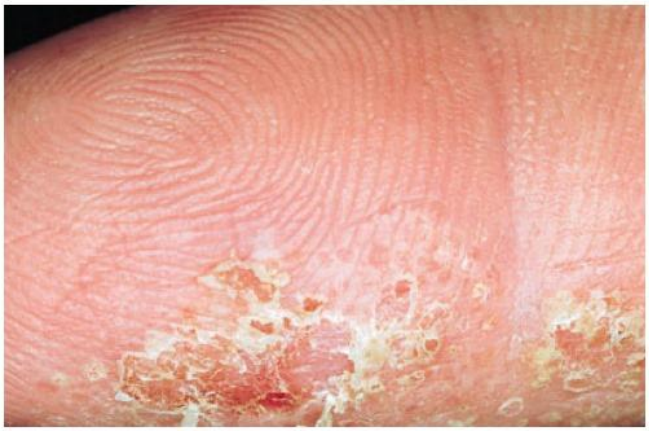


Appendix D



Appendix E

Predicted: 86, True: 6



Predicted: 86, True: 25

Predicted: 86, True: 48