Gender Classification Based on Hand Images

David Dickson
School of Electrical and Computer Engineering
Georgia Institute of Technology
ddickson30@gatech.edu

Nathan Wang
School of Electrical and Computer Engineering
Georgia Institute of Technology
nywang08@gatech.edu

Damian Huerta
School of Electrical and Computer Engineering
Georgia Institute of Technology
damhue039@gatech.edu

Laurence Leon
School of Electrical and Computer Engineering
Georgia Institute of Technology
lleon30@gatech.edu

***Abstract*—This paper seeks to find a better image classification model for classifying gender based on images of hands. Previous work has been done on the data set using an SVM and CNN model. This model was complex, using two CNNs and a combination of four SVMs to predict gender based on images of hands. The group hoped to reduce the complexity of the model while achieving similar or better model accuracy performance. Additionally, this paper focuses on resizing images and the effect that they have on model accuracy and training. A thorough investigation of CNNs and in particular the VGG-16 model was done to see what improvements could be made by varying model parameters such as dropout, number of hidden layers, different activation layers, and different optimizers. These results are compiled and will be reflected in this paper.**

***Index Terms*—Convolutional Neural Networks (CNNs), VGG-16, Optimization**

I. Introduction

For this project, the team wanted to explore refining and improving CNN based image classification algorithms. With members on the team interested in autonomous vehicles and developing camera systems to recognize humans, learning more about CNNs and their implementation was of interest to the team. This is what lead to the topic of refining a current implementation to reduce the complexity of the model as well as maintain or improve model accuracy. The original model developed for the project used two CNNs and a combination of four SVMs to train the model [1], [2]. The team wanted to use one CNN and see if they could achieve the same or better results by tuning different model parameters. These model parameters include dropout, number of hidden layers, activation layer, and optimizers.

The team would also find out that they would need to reduce the image size of the dataset, as the current image size required too many GPU resources. This led to an additional investigation during the experimentation to determine if a reduction in image size and resolution would reduce the model validation accuracy.

II. Methodology

The data set used in this project consisted of roughly eleven thousand samples with nearly four thousand male samples and around seven thousand female samples. Originally, each sample had a generous resolution of 1600x1200 which required a substantial amount of computation. Within the data set, there was a mixture of images with hands in pronated and supinated orientations along with variations in location, angles, and rotations. Nine examples from the data set are shown in Fig. 1.

Fig. 1. Sample Hands used for Training and Validation

As the goal was to find a CNN that excelled in image classification, multiple architectures of varying hidden layer size, perceptron amount, optimizers, and activation layers were researched and compared. The most notable architectures analyzed were ResNet, AlexNet, VGG-16, and Inception v1/GoogLeNet.

After weighing the advantages and disadvantages of each model, the team finally decided on VGG-16. AlexNet had the least hidden layers out of the 4 primary architectures, with a total of 8 layers [3]. Since having too few hidden layers could result in underfitting, AlexNet was rejected. On the other hand, the architecture ResNet had too many hidden layers, with 152 layers. With too many hidden layers, not only could overfitting occur, but overall complexity of the model would increase which would increase the run-time and storage significantly. The reasoning behind choosing VGG-16 over GoogLeNet was due to a close to negligible difference in overall accuracy and that VGG-16 had less hidden layers than GoogLeNet, with 16 layers compared to 22 layers respectively. The VGG-16 architecture is visualized in Fig. 2.

Fig. 2. Model Structure of the VGG-16 Architecture

In order to combat the compute issue with the original size of the images, the team chose to down-sample our input from 1600x1200 to 224x224. This would allow the team maximize the use of the run-time memory storage. The resizing process was achieved by using python's PIL library to resize all images to various square resolutions used during testing. Luckily, most images had abundant white space around the hand, so resizing rarely lost any hand-content in the image.

For experimentation of the model, the data was split into a 20-80 split with 20 percent of the data being used for testing and 80 percent of the data being used for training.

III. Results/Discussions/Findings

The team, as previously mentioned, was interested in improving upon the original algorithm. The original algorithm used two CNNs and four SVM classifiers in combination to produce the end model that achieved an overall validation accuracy of 97.5 percent [1]. The team's goal was to reduce the model complexity and resources required but still achieve high validation accuracy. After completing the research, the team found that it would be best to use the CNN model VGG-16, which is commonly used for image classification [4].

After developing the unmodified version of the VGG-16 model, which is explained further in row two of Table I, the team arrived at a 95.53 percent model validation accuracy. While this was still a high accuracy, the team wanted to achieve at least a similar validation accuracy to the original model. Initially, the team was unsure if they would be able to achieve a result similar to or better than the original model due to modifications to the original image size, described earlier. This uncertainty was because size modifications could result in resolution loss that could affect the accuracy of the model generated. With this in mind, the team continued with the project to see if improvements could be made. Table I lists the key results of the team's model variations. Each model variation will be discussed in the following sections.

TABLE I
Summary of Model Accuracy for Sampled Model Parameters

Accuracy (%)	**Dropout**	**Hidden Layers**	**Activation Layer**	**Optimizer**
95.53	0.0	16	ReLu	ADAM
94.04	0.0	16	ReLu	SGD
63.52	0.0	16	Sigmoid	ADAM
96.93	0.0	3	ReLu	ADAM
98.01	0.2	16	ReLu	ADAM
96.50	0.8	16	ReLu	ADAM

A. Dropout

The team varied the probability of the dropout layers that comprise the final decimation layers of the VGG-16 architecture. The dropout tests can be summarized by rows six and seven of I. For a dropout probability of 0.2, a model validation accuracy of 98.01 percent was achieved, while a dropout probability of 0.8 resulted in an accuracy of 96.50 percent. The accuracy of the model for both of these dropout probabilities during training is shown in Fig. 3 and Fig. 4.

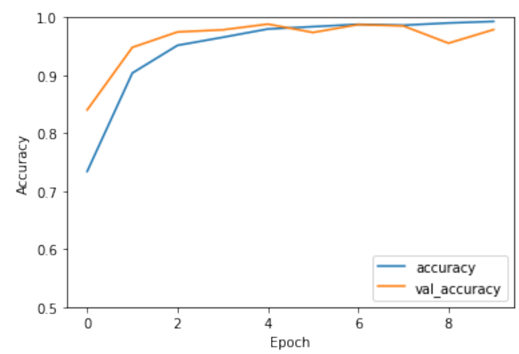

Fig. 3. Validation Accuracy Using A Dropout Probability of 0.2

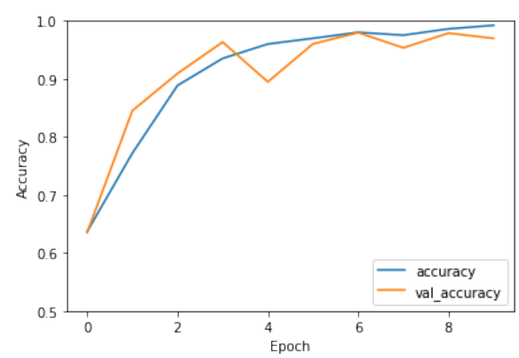

Fig. 4. Validation Accuracy Using A Dropout Probability of 0.8

B. Hidden Layers

The team reduced the number of hidden layers present in the base VGG-16 architecture from 16 to 3, effectively creating a VGG-3 architecture. This was in an effort to test whether a large reduction in model complexity had a significant positive or negative effect on validation accuracy. Row five of I shows the model setup for this test. This model resulted in a model validation accuracy of 96.93 percent. The training of this model is shown in Fig. 5. It is important to note here that this model did provide a higher model validation accuracy compared to the base model. While there is a higher model validation accuracy, it will be seen later that through refinement of the VGG-16 model, the model validation accuracy will be improved over the VGG-3 architecture.

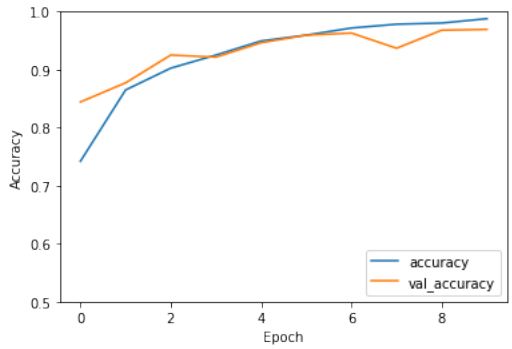

Fig. 5. Validation Accuracy Using Three Hidden Layers

C. Optimizers

The team varied the optimizers for the model to see if any performance improvement could be achieved on top of the base VGG-16 model. Here, the team tried ADAM and stochastic gradient descent (SGD) optimizers. While both optimizers achieved high model validation accuracy, with ADAM achieving 95.53 percent and SGD achieving 94.04 percent, ADAM produced the best results that were similar to the base VGG-16 model, which also used ADAM for the optimizer. Plots of these results can be seen in Fig. 6 and Fig. 7, with Fig. 6 showing the validation accuracy using ADAM for the optimizer and Fig. 7 showing the validation accuracy using SGD for the optimizer.

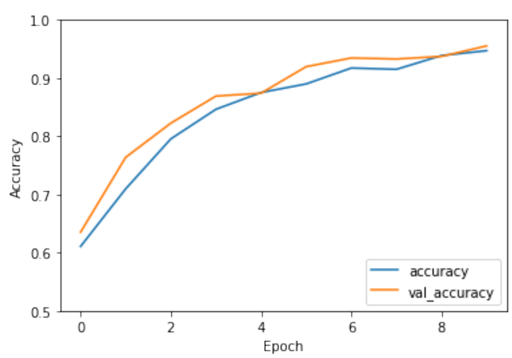

Fig. 6. Validation Accuracy Using ADAM As Optimizer

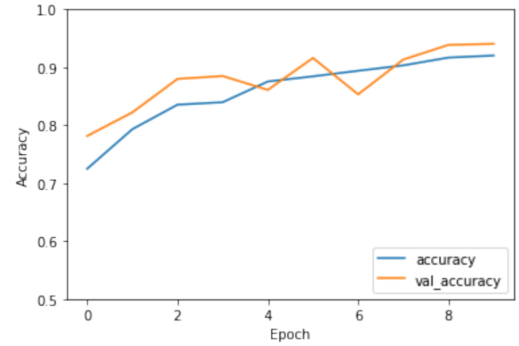

Fig. 7. Validation Accuracy Using SGD As Optimizer

D. Activation Layers

The team varied the activation layers of the baseline model. Here, the team tried ReLu and sigmoid for the activation layers. These two activation layers produced drastically different results. When using ReLu as the activation layer, the model achieved 95.53 percent accuracy, but when using sigmoid as the activation layer, the model validation accuracy fell to 63.52 percent as seen in Fig. 8.

The team believes this is due to vanishing gradients when using sigmoid as the activation layer. In this case, the kernel weights would have a distribution near zero. The team then plotted these weights and found that they were converging toward zero as the network layers got deeper as seen in Fig. 9. This provided confirmation to the team that sigmoid was suffering from vanishing gradients in this case.

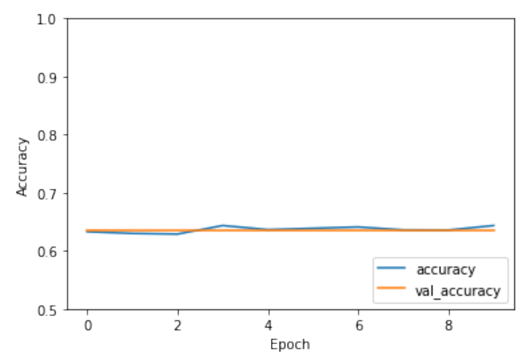

Fig. 8. Validation Accuracy Using Sigmoid As Activation Layer

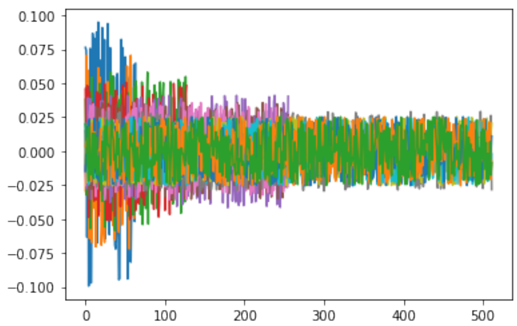

Fig. 9. Model Weights Per Layer

IV. Conclusions

The team set out to test how a commonly used image classification learning model, VGG-16, could be tuned via standard methods discussed in the course to achieve improved performance in the specific application of gender classification given hand images. These methods included dropout extent, number of hidden layers, optimizer choice, and the type of activation layer. The performance metric is based on the model validation accuracy.

Our results indicate that the best VGG-16 based model for this classification task had a 0.2 dropout probability, 16 hidden layers, ADAM optimizer, and ReLu activation function. This model achieved our best accuracy result: 98.01 percent. If model complexity is of major importance, a model with 0 dropout probability, three hidden layers, ADAM optimizer, and ReLu activation layer model would also be a viable option. This model has significantly fewer parameters to train and evaluate than the VGG-16 model, while maintaining a validation accuracy of 96.93 percent.

Additionally, the team found that while they did have to resize the images in the original data set, this did not greatly compromise the model validation accuracy. While there was a slight reduction in accuracy this was made up for in the refinements of the model.

Through the experiments, the model created by the team, which used only a CNN model, performed just as well, if not better, than the original model, which used SVMs and CNNs. These results indicate that a simpler model could be equally viable which would result in quicker runtime and less space required.

In the future, work could focus on pursuing the best models found in this study. Models that are like VGG-3 but incorporate dropout and different activation layers could find a better model in terms of accuracy. Also, future work could focus on model complexity, training run-time, and evaluation run-time to give a better picture of the trade-offs between validation accuracy and different time investments.

Team Member Contributions

A. David Dickson

David Dickson worked on the base model development using VGG-16. He also helped to complete some of the model experiments as well as investigate the issues related to using sigmoid for the activation layer in the model. Additionally he helped to write the abstract, results, and introduction sections.

B. Nathan Wang

Nathan Wang was responsible for the research of the initial model, which resulted in the selection of the initial VGG-16 architecture. He also assisted in the development and debugging of the model. Additionally, he helped write the methodology and the introduction/conclusion.

C. Damian Huerta

Damian Huerta worked on the establishment of the base model and development of various models to test our setup process. He also participated in various model experiments changing things such as layer configurations, activation functions, optimizers, and image sizes. Damian was also in charge of creating a python script to sort and resize all of the data samples used to maximize run-time memory usage. Finally, he helped write the methodology section.

D. Laurence Leon

Laurence Leon was responsible for running the bulk of the model variation tests, generating the model accuracy results and figures, and sending them to the rest of the team. He also helped debug the VGG-16 architecture when suspicious results were being found early on. Finally, he wrote the dropout, hidden layers, and conclusions sections of this report, as well as contributing to overall editing and reference tracking.

References

[1] M. Afifi, "11K hands: Gender recognition and biometric identification using a large dataset of hand images," Multimedia Tools and Applications, vol. 78, no. 15, pp. 20835–20854, 2019.
[2] Kanchan T, Krishan K, "Anthropometry of hand in sex determination of dismembered remains-a review of literature," J Forensic Leg Med 18(1), pp. 14–17, 2011
[3] Karim, Raimi, "Illustrated: 10 CNN Architectures," Retrieved from https://towardsdatascience.com/illustrated-10-cnn-architectures-95d78ace614d
[4] Thite, Avinash, "Introduction to VGG16 — What is VGG16," Retrieved from https://www.mygreatlearning.com/blog/introduction-to-vgg16/