AASD4015 ADVANCED MATHEMATICAL CONCEPTS FOR DEEP LEARNING

GlassCheck: An Automated System for Detecting Eyewear Presence

Group – 8

Hetvi Raseshkumar Shah – 101439231

Yash Upadhyay – 101378657

Vivek Patel - 101432083

Vishal Patel – 101446059

TABLE OF CONTENTS

1. ABSTRACT

In this report, we address the problem of glasses detection in images, aiming to determine whether a person is wearing glasses or not. We utilize deep learning techniques, specifically fine-tuning a pre-trained MobileNetV2 model, for this purpose. The dataset comprises images of individuals both with and without glasses, which are preprocessed and split into training, validation, and test sets.

Our approach involves freezing and selectively unfreezing layers to adapt the model to the glasses detection task. We analyzed the effects of fine-tuning on accuracy and loss metrics during training and validation. Furthermore, predictions are made on the test dataset, and visual comparisons between predicted and actual labels are presented.

Results indicate that fine-tuning enhances model accuracy and generalization for glasses detection. This report provides a detailed overview of the methodology, presents results, and discusses implications for image classification and object detection tasks.

2. INTRODUCTION

2.1 Problem Description: We tackle the task of identifying whether people in images are wearing glasses. This problem is vital in various domains, from security to user authentication. Our approach involves using deep learning to teach computers this skill.

2.2 Importance: Being able to spot glasses matters for security cameras, accurate user verification, and targeted marketing. We collected images, used a smart model, and fine-tuned it to detect glasses.

2.3 Overview of Results: Our results show that our method works well. The model became better at spotting glasses after fine-tuning. We discuss our process, share our findings, and explore how this can help computers understand images better.

3. RELATED WORK

Others have also been working on the challenge of spotting glasses in pictures, which has important uses. They've used smart computer programs and similar techniques, like the MobileNetV2 model we're using.

What sets us apart is how we fine-tune the model. We focus on making it better at recognizing glasses by adjusting specific parts. This part is new and makes our approach different. We test and show that this fine-tuning improves the model's accuracy.

While we share some ideas with others, our unique focus on fine-tuning makes our work stand out. We'll explain more about how we do it and what we find out in the next sections.

4. ABOUT DATA

For our project, we worked with a specialized dataset obtained from Kaggle. The dataset, available at https://www.kaggle.com/datasets/jeffheaton/glasses-or-no-glasses contains images of individuals with and without glasses. It is a curated collection of pictures sourced from various real-world scenarios.

We chose this dataset due to its relevance to our problem and the availability of labeled examples. While the original dataset contains a substantial number of images, we focused on a subset to streamline our experimentation process.

To prepare the dataset for our glasses detection task, we performed preprocessing steps, including resizing the images to a uniform size and categorizing them based on the presence or absence of glasses. This organized dataset served as the foundation for training, validation, and testing our deep learning model.

5. METHODOLOGY

5.1 Data Preprocessing

Getting the photographs ready, for the analysis was our first step. To ensure consistency, we made sure each image was 160 pixels by 160 pixels.
After that, we divided the images into two categories: people wearing glasses and people without spectacles. The computer was able to recognise glasses because to this sorting.

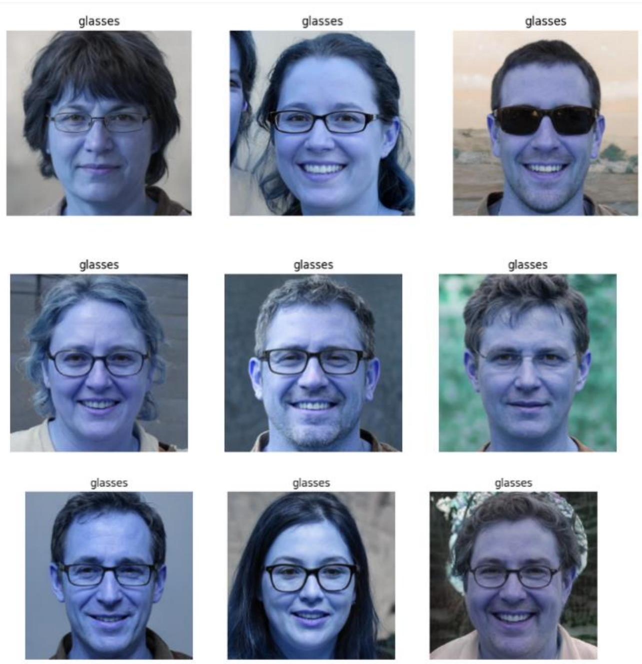

Figure 5.1.1 Images with glasses

Figure 5.1.2 Images without glasses

5.2 Train, Test and Validate

We divided our preprocessed data into three sets: training, testing, and validation. These sets helped us fine-tune our model, check its learning, and evaluate its performance.

We divided the data into 70:30 split for training and validation set, which include 3444 and 1476 images into each class respectively. Then we obtained testing batch from validation batch by dividing validation batch with 5. So finally, we have 108, 38, and 9 for our training, validation, and testing batch sizes respectively. As we were preprocessing the dataset, we found out that our data is imbalance, we have more images with glasses then without glasses.

5.3 Data Augmentation

Now we have performed data augmentation task with training dataset of the images. For this method we have augmented our dataset horizontally with 0.2-degree rotation in random direction, which means the images were rotated by 0.2 degree in randomly chosen direction as shown below.

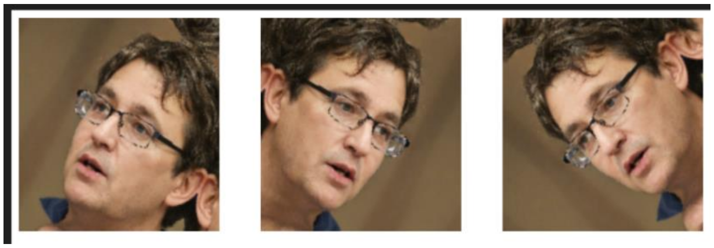

Figure 5.3.1 Augmented Images

5.4 Base Model for Transfer Learning and Rescaling Pixel Value

First, we have rescaled the pixel values between -1 and 1, then we have performed transfer learning task by using MobileNetV2 pretrained model which is developed by Google containing a dataset of 1.4 million images with 1000 classes. MobilenetV2 is CNN architecture based on inverted residual structure where residual connections are between the bottleneck layers.

Now, created our base model with MobileNetV2 where we have not included our top classification layer for this model since it has been already added when previously classification layer created.

5.5 Feature Extraction

Now, we extracted the features from our base model that created using MobileNetV2. In order to complete this task, we need to freeze the convolutional base from our basic model because that base will be used as feature extractor. This feature extractor will convert the image from 160x160x3 to 5x5x1280. Then we have added classification head to convert the feature in 1280-element vector per each image and dense layer for single prediction.

5.6 Combined & Compile the model.

In this method, we have combined all the above process to our model and get the summary of the model which shows that our model has about 22 million parameters. Then we have compiled the model using Adam optimizer and used Binary Cross entropy loss function to measure loss in the model. Finally, our model is ready to train.

5.7 Fine Tuning

We used fine tuning to our model to increase the performance with layers of our based model. For this task first we have unfrozen our top layer from base model then applied weights to the models which

were not updated during initial training of our model. Initially, the model had 154 layers, but for this task we have trained our model on 120 layers.

6. INITIAL RESULTS

6.1 Base Model results

Based on our methods mentioned earlier, we have collected the training and validation accuracy with their loss of our models. So, as we mentioned above, we have trained our model twice. First, we have used the base model with all the processes and trained the model which gave us the testing accuracy of 0.93 and testing loss is about 0.247. As shown in the below figure that the accuracy is increasing, and loss is decreasing with satisfying rate with epochs.

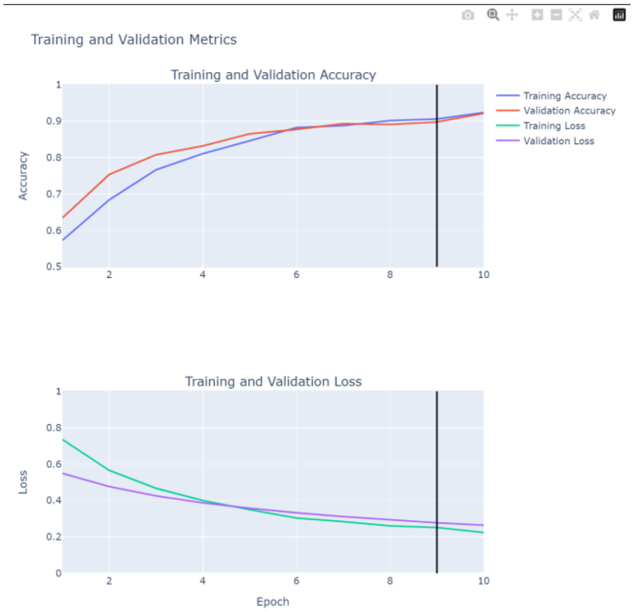

Figure 6.1.1 Graph of initial model

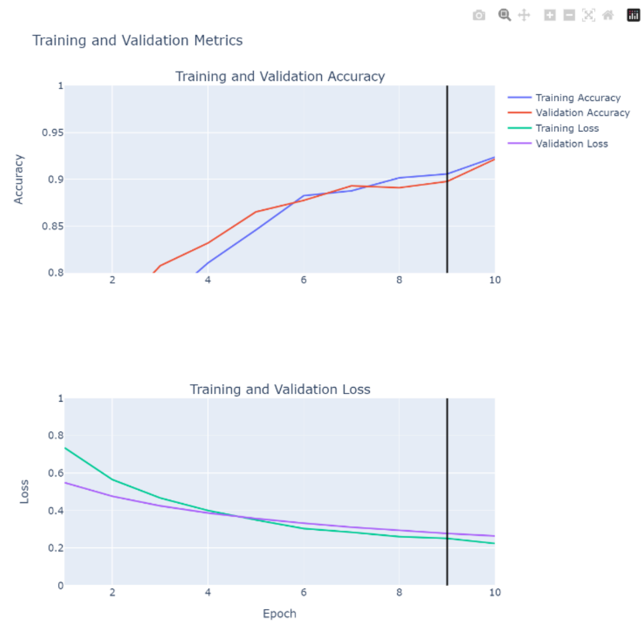

Figure 6.2.1 Graph of Fine-Tuned Model

6.2 Fine-Tuned Model results

Then we performed fine tuning on the same model with 120 layers, which was initially had 154 layers, end up giving more accuracy than initial model. We obtained testing accuracy of 0.94 and testing loss of 0.19 which is better than initial model. As shown in the below figure, the loss does not change a much but when we look at the accuracy of the model, we can clearly see the change as epochs are increasing and we can see that after fine tuning model has better performance.

And finally, below figure shows the images with its label, glasses or no glasses.

Figure 6.2.2 Results of testing images with labels

7. EXPERIMENTS

To prove that our approach to the problem solve the issue, we conducted an experiment with testing images and real time images. First, we used testing set of the images from dataset and the results are discussed earlier.

7.1 Data

To make the project more interesting, we collected data from our friends with different poses of their face with different filters and something wearing other than glasses. For example, some of the images are without glasses and plain, filters and mask or hat, some images are with glasses with filters, plain and hat.

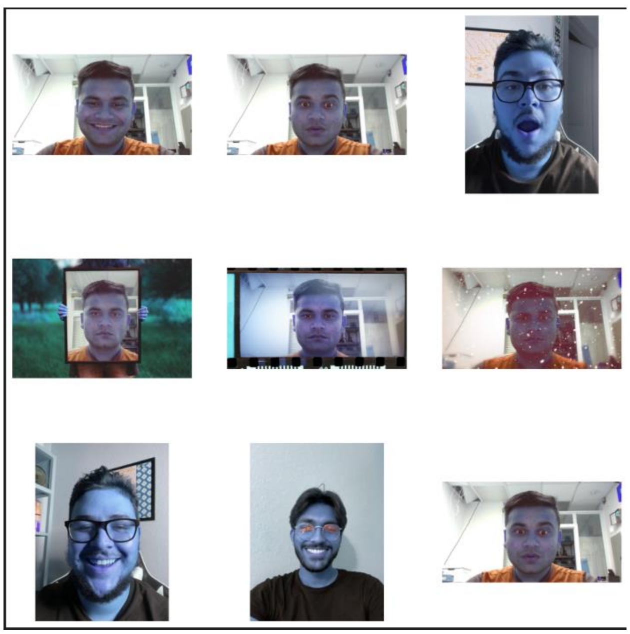

Figure 7.1.1 Real Time Images

7.2 Results

Since our training dataset does not contain images with filters, mask, or hat, we expected to get a bit lower accuracy than our testing dataset but not lower than 60%. We obtained the accuracy of 62% for our real time dataset and the output with its label shown in the figure 7.2.

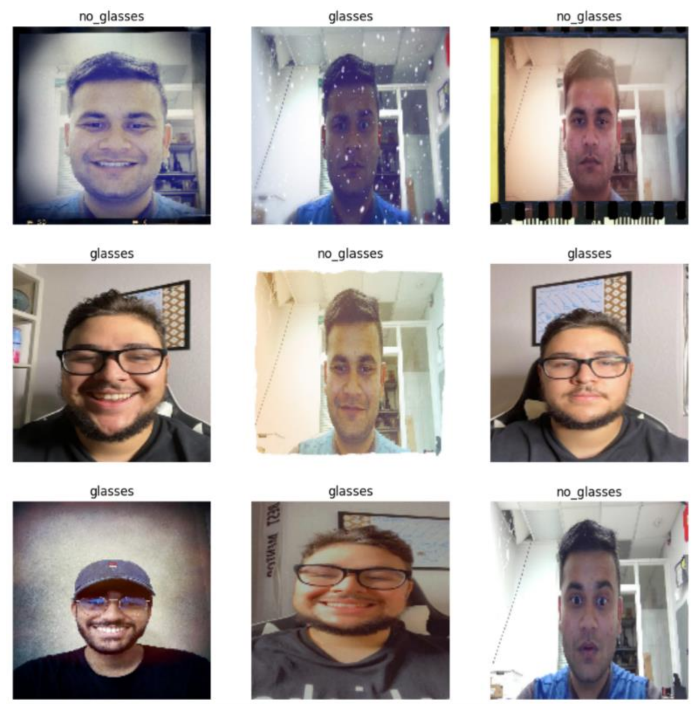

Figure 7.2.1 Results of real time images with their output label

8. CONCLUSION

In this project, we created an automated system named "GlassCheck" that effectively tackles the task of automated eyewear presence detection using advanced deep learning techniques. Through the utilization of a fine-tuned MobileNetV2 model, we achieved significant enhancements in both accuracy and overall performance. Our systematic methodology commenced with thorough data preprocessing and augmentation to establish a well-balanced dataset. By leveraging transfer learning with the MobileNetV2 architecture, we harnessed its robust attributes for precise feature extraction. Notably, the process of fine-tuning emerged as a crucial stage, profoundly elevating the model's ability to accurately identify the presence of glasses. This was substantiated by an impressive 94% testing accuracy and a testing loss of 0.19, highlighting the pivotal role of fine-tuning in adapting pre-existing models for specific tasks.

Furthermore, our project expanded its scope beyond the initial dataset by incorporating real-time images that encompassed diverse real-world scenarios. Despite encountering challenges posed by various elements such as filters, masks, and accessories not encountered in the training data, our model demonstrated remarkable adaptability, achieving a commendable accuracy of 62%. The GlassCheck system presents a robust and practical solution for automated glasses detection, with potential applications spanning security, user authentication, and marketing domains. Our comprehensive approach, encompassing meticulous data preprocessing, effective transfer learning, and impactful fine-tuning, effectively showcases the potency of deep learning in resolving intricate image classification challenges. As we progress, continued research and refinement hold the promise of further enhancing the model's capabilities and effectiveness across a wide range of complex scenarios.

References

ADVANCED APPLIED MATHEMATICAL CONCEPTS FOR DEEP LEARNING CRN-82362-202203, In class notebooks, Prof. Ran Feldes

https://www.kaggle.com/datasets/jeffheaton/glasses-or-no-glasses

https://www.kaggle.com/code/aepeters/ensemble-model-using-functional-api

https://keras.io/api/applications/