Yassir JALAL, Vicky WENG
June 6th, 2024

ICPBL Project Report

Introduction :

In the Introduction to Artificial Intelligence course we have to do a project. A project that is particularly interesting and important due to recent situations of the Covid-19. This project consists of a mask detection program that should tell us if the person on a provided photo wears a mask or not. With those tasks, we created a deep learning model that can classify whether the person is wearing a mask or not. Overcoming the challenge of limited masked face datasets through synthetic data generation using the MaskTheFace tool,we implemented a Resnet-50 architecture for binary classification. The project was executed on Google Colab, following a structured workflow. Thanks to the model's code, we can see its accuracy and reliability in real-world scenarios.

MaskTheFace tool :

MaskTheFace is a computer vision script that masks faces in images. It uses a dlib-based face landmarks detector to find faces and six key facial features needed to apply a mask. Based on the face tilt, it selects a corresponding mask template from a library. Then, it adjusts the template to fit perfectly on the face using the six key features. MaskTheFace offers a variety of masks to choose from. Instead of gathering a dataset of masked faces, it can convert existing face datasets into masked-face datasets. It identifies all faces in an image and applies the chosen masks, considering factors like face angle, fit, and lighting conditions. You can input a single image or a whole directory of images into the code.

A big advantage with this tool is that it uses different types of masks which is good because it helps the model to be more accurate since there is not just one type of mask in the real world. For example :

empty

gas

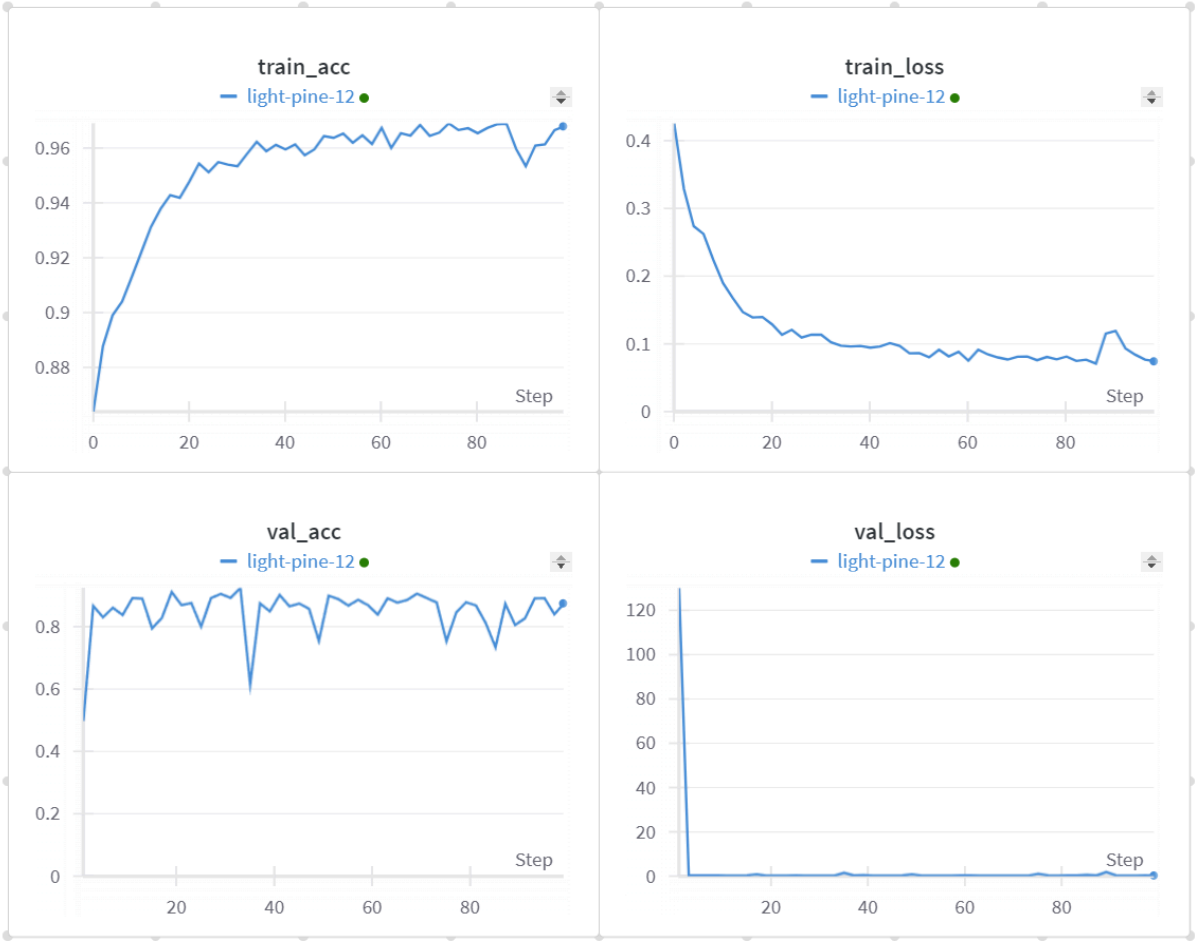
N95

cloth

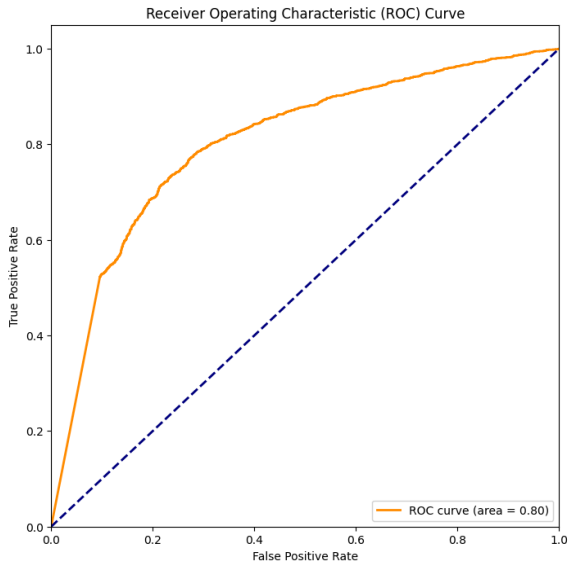
Data augmentation :

We also added random augmentation to the masked faces, like random flip, rotation, and coloring modifications such as brightness, contrast and saturation modifications.

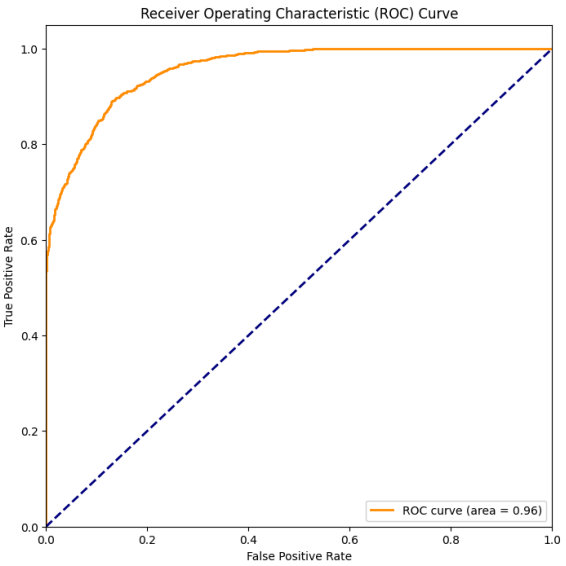
Plots :



ROC curve and AUC :

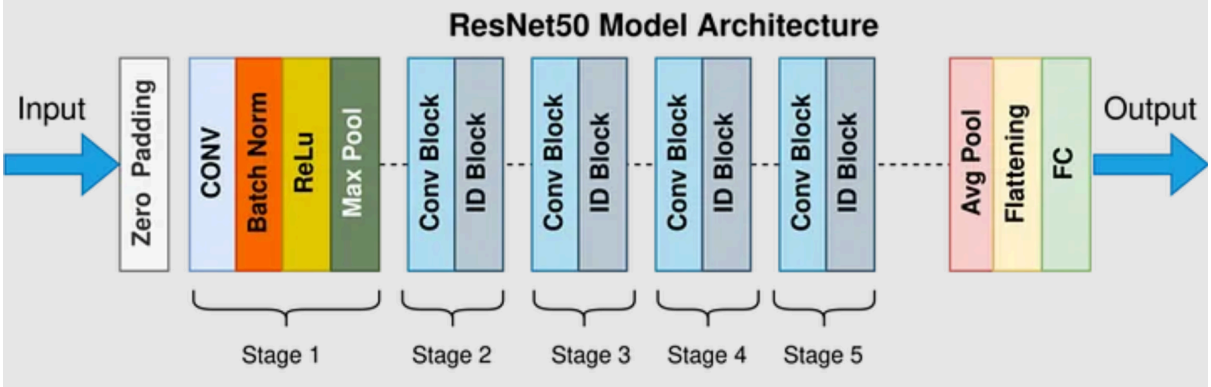


Epoch 1



Epoch 50

Diagram of the Resnet-50 architecture



ResNet-50, comprising 50 layers divided into 5 blocks with residual blocks, preserves information from earlier layers to enhance input data representation. It starts with a convolutional layer performing convolution on the input image, followed by max-pooling to downsample the output, then passes through residual blocks. Each residual block has two convolutional layers, each with batch normalization and ReLU activation, and adds the second layer's output to the block's input, followed by another ReLU activation. The network's final layer is fully connected, mapping the last residual block's output to the output classes, with neurons corresponding to the number of output classes.

Discussions :

- Changing hyperparameters (learning rate, batch size, etc.)

Changing hyperparameters can really affect the performance and training of one's model. There are many hyperparameters that are involved such as the learning rate, the batch size, the optimizer, or even the number of epochs.

If the learning rate is set too high, the model's weight could stay around the minimum of the loss function or even diverge. This could cause a really poor convergence. On the contrary, if the learning rate is too low, the model will converge very slowly. This can mean long training times and it could get stuck in a local minima.

There is also a big difference between a small batch size and a big one. Having a small or big batch size can affect the noise during the training which can lead to a slower training process or an overfitting of the model.

The number of epochs used is also really important and can have a big impact on the results. With our model, if we did a training with 10 epochs, the train and val accuracy were respectively 0.9419 and 0.9111 whereas if we did it with 50 epochs, they were 0.9679 and 0.8747

Overall, in order to have the best model is to know which hyperparameters are more appropriate and that is by testing and running the code over and over again.

- Changing model architecture (Resnet 152, etc.)

Using a model architecture like ResNet152 can enhance our project due to its increased depth and capacity to learn more complex features. ResNet152 has 152 layers compared to ResNet50's 50 layers, allowing it to capture finer details and more intricate patterns in the data. This deeper network can model more complex relationships, potentially improving accuracy and performance on tasks requiring high levels of feature extraction.

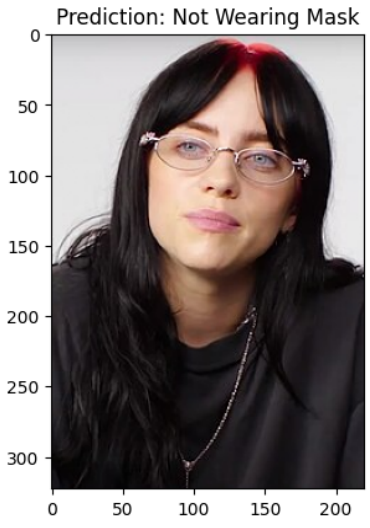
Additionally, ResNet152 benefits from the same residual connections that help prevent vanishing gradient issues, enabling effective training even with its increased depth. This allows ResNet152 to leverage its greater number of layers without suffering from the training difficulties typically associated with very deep networks. As a result, using ResNet152 can lead to improved performance, especially on more complex datasets where the extra capacity of the deeper network can be fully utilized.

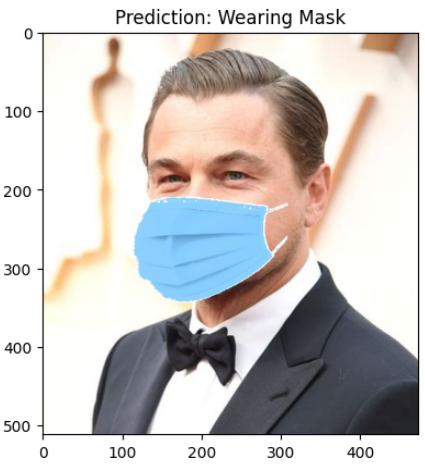
Limitation of the model :

Our model is far from being perfect, and that has to do with many limitations that we have.
First of all we have some data quantity limitations. The dataset we used contains 6000 photos. It can seem sufficient but the more photos we have to train our model, the better. So the model could be more effective if we used more photos but that would also mean a lot more time of training.
Then, we have the model architecture limitations that tell us if the model is overfitting or underfitting. We believe that our model is a little bit overfitting but not much because it performs well on the training set and a little worse on the validation set. In general, the train accuracy is around 0.96 whereas the val accuracy is around 0.84.
Finally, we had difficulties setting up the model and coding it, so we did not have much time to test it. Due to the long training duration, we didn't test it many times so with more time, we could have maybe changed it and perfected it.

Qualitative evaluation:

Here are some celebrities we tested with our model :





Difficulties :

The MaskTheFace tool was difficult to get around at first because it was our first time using it but once it worked, it was easy.

The tool mostly works but there were still some issues. Sometimes, the mask wasn’t put on the right person, and many times, it just did something weird around the person’s mouth with the "inpaint” mask, which isn’t really representative of reality.

Mask not on the right person

Example of an “inpaint’ type mask