Udacity Machine Learning Engineer Capstone Project

Face Mask Detection using CNN

Hiep Doan

Dec 30, 2021

Domain Background

Since the beginning of 2020, Covid-19 has brought the world to a halt. Starting in Wuhan in December 2019, the coronavirus spread worldwide rapidly. At the time of writing, 281 million people had gotten Covid-19, and 5.4 million people died. In the fight against the novel virus, many preventative measures, including social distancing, masking, and lockdowns, were set in place. Wearing a face mask remains one of the most effective ways to prevent the spread of Covid-19 in public areas. In addition, face mask detection is an excellent application for facial attribute classification problems.

Problem statement

The project's goal is to build a classifier that helps to ensure a person is wearing a mask correctly. Given an image of a person in JPEG format, the system tells whether a person is wearing a mask and whether the mask is worn properly.

Datasets and inputs

The face mask dataset (https://www.kaggle.com/andrewmvd/face-mask-detection) contains 853 images in PNG format with three classes: with a mask, without a mask, and mask worn incorrectly.

There is a corresponding annotation file in XML format for each image. The annotation contains metadata about the image and bounding boxes in PASCAL VOC format. In addition, each bounding box is annotated with a class.

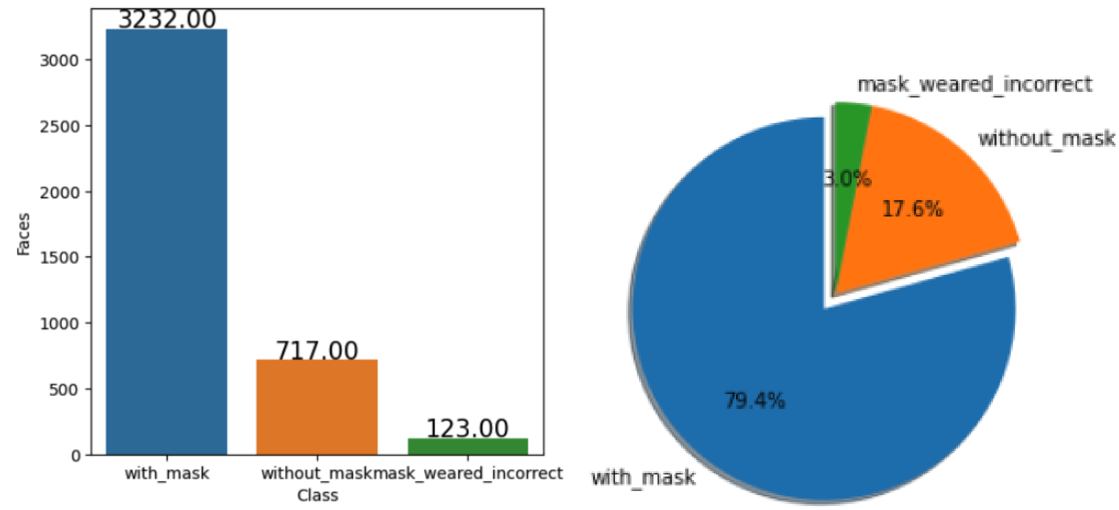


Figure 1. Data Distribution

Figure 1 shows the distribution for this dataset. There are 4072 annotated faces in these images. Out of which, 3232 (79.4%) are with masks, 717 (17.6%) are without masks, and 123 (3%) are with incorrectly worn masks.

Solution Statement

Convolutional Neural Network (CNN) has proved to be one of the best algorithms to classify facial attributes. In this project, we train a CNN model to perform binary classification. Faces are extracted from original images and cropped to the same size. They are then split into training and testing images. The model is trained with training images and tested against testing images.

Benchmark Model

	Actual		
Predict		**with_mask**	**without_mask**
	with_mask	True Positive (TP)	False Positive (FP)
	without_mask	False Negative (FN)	True Negative (TN)

Figure 2. Confusion Matrix

Precision = $\frac{TP}{TP + FP}$

Recall = $\frac{TP}{TP + FN}$

F1-score = $2 * \frac{Precision * Recall}{Precision + Recall}$

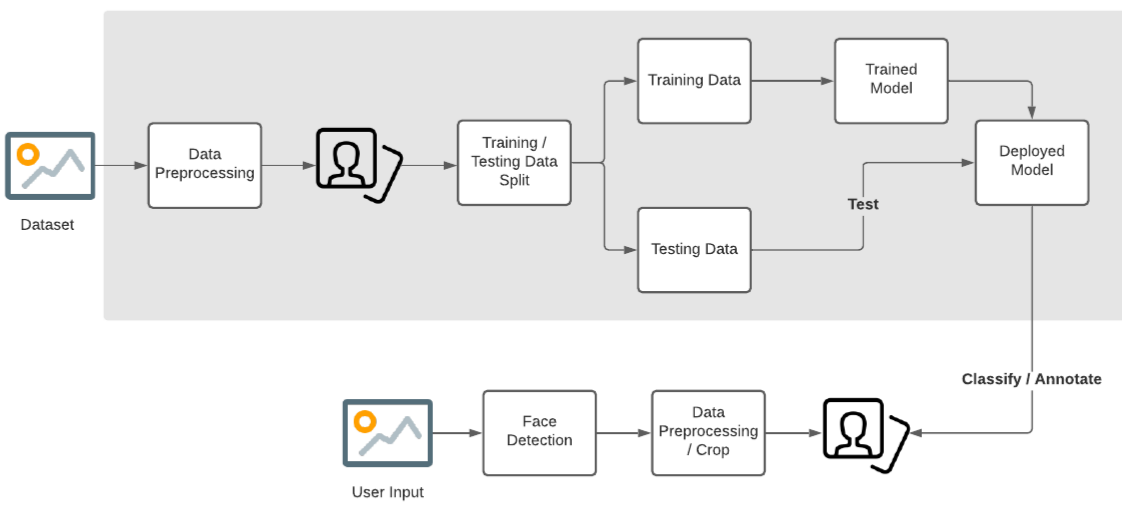
Since the cost of misclassifying someone without masks is high, we want to reduce our False Positive rate. F1-score is a better metric than accuracy in this case, especially when our data distribution is skewed towards the with_mask class.

A benchmark of 80% for f1-score is used in this project.

Evaluation Metrics

The project uses f1-score as the evaluation metric.

Project Design



Data Preprocessing

We perform the necessary steps to prepare our data.

- Transform XML annotation files to CSV format for training and testing.

- Crop and stretch bounding boxes of faces to the same size and save them to new images.

Data Split

We split data into a training set (80%) and a testing set (20%)

Model Training

We start with a simple CNN and iteratively tune our hyperparameters until a good f1-score.

Model Deployment and Testing

We deploy our model and test the model against the testing dataset.

Application

We use OpenCV pre-trained cascade classifier for images from users to detect faces and process these faces. We then use our model to perform classification and annotate the results on original images.