Face Mask Detection

Udacity Machine Learning Engineer Nanodegree Capstone Proposal

DaeMyung Kang
August 30, 2020

Project Overview

This document is a proposal for the Capstone Project of the Udacity Machine Learning Engineer Nanodegree Program. The goal of the capstone project is to create a Deep Learning Model that detect a person is wearing a mask.

Domain Background

COVID-19 is very serious problem in the world now. Wearing mask is most helpful to prevent the spread of COVID-19.[1] Many Countries are compelled to wear masks for their people. Mayor of Gyeonggi-do in South Korea orders that wear a mask. If someone violate it, its fine is almost $2500.

It spends much efforts to detect someone that don't wear a mask In the crowdy subways or buses. To automatic detect the faces to wear a mask is very helpful to prevent spread the COVID-19.

Problem Statement

This is a binary classification problem. Inputs are image of peoples, and the goal is to detect peoples wear a mask or not. Given an image of some peoples, the model will detect the persons to wear a mask or not.

But It needs face detection from images. I will use MTCNN to detect face from images.

Datasets and Inputs

Kaggle Datasets link is here.(

https://www.kaggle.com/ashishjangra27/face-mask-12k-images-dataset

The datasets have only face images and tag as a directory name. there are 3 main directories and each one has two tag directories.

		Count
Train	WithMask	5000
	WithoutMask	5000
Test	WithMask	483
	WithoutMask	509
Validataion	WithMask	400
	WithoutMask	400

The images size are diverse. Below images are each sample for WithMask and WithoutMask.

[WithMask]

[WithoutMask]

These datasets are almost even images for WithMask and WithoutMaks.

Solution Statement

This is a supervised learning problem. The dataset is all labeled. The goal is to determine that people wear a mask or not. It is a kind of image classification. Deep Learning is known as the best solution for image classification problems. Using Convolution neural networks[2] will be a good first step for this.

In Image classification category, we can use "Transfer Learning"[3], and "Transfer Leaning" shows

better performance compared to train model from scratch. There are many good pretrained models for this.(ex, resnet, mobilenet, vgg, inception)

The solution will follow below steps.

- Create baseline Model to use Simple Convolution neural networks to set benchmark model.
- Using "Transfer Learning" to get good baseline performance than benchmark model.
- Tune final model performance

Benchmark model

The benchmark model will be a simple CNN classifier trained on the train data. Then I will try to improve the model performance by adding more layers and user this model as a benchmark to test performance.(Generally, using more deep layers show better performance), and I will use "Transfer Learning" using Resnet as base pretrained model as final model.

Evaluation Metrics

I will use f1-score[4] as Evaluation Metric to quantify the performance of booth the benchmark model and the solution model.

	True Condition Positive	**True Condition Negative**
Predicted Condition Positive	TP = True Positive	FP = False Positive
Predicted Condition Negative	FN = False Negative	TN = True Negative

Accuracy, Precision and Recall can be derived from the above matrix.

- Accuracy = (TP + TN)/(TP + FP + TN + FN)
- Precision = TP / (TP + FP)
- Recall = TP / (TP + FN)
- F1_SCORE = 2 * (Precision * Recall)/(Precision + Recall)

Project Design

- Programming Languages and Libraries
  - Python 3
  - Jupyter notebook
  - Pytorch, numpy, matplotlib
- Data Exploration and Visualization
  - View Images
  - Basic statistics and understandinf of the dataset
- Data Processing
  - Basic cleaning or Preprocessing if needed
  - Split the data into train, test, validation sets
- Build Benchmark CNN Model
  - Implement simple CNN model
  - Training and Testing
  - Increasing more layers.
- Build Transfer learning model
  - Choose Pretrained Model(ex, resnet, mobilenet)
  - Implement transfer learning CNN Model
  - Training and Testing
- Model Tuning
  - Fine tune the model's hyperparameter
- Testing

Reference

[1] https://www.the-scientist.com/news-opinion/how-face-masks-can-help-prevent-the-spread-of-

covid-19-67646

[2] https://cs231n.github.io/convolutional-networks/

[3] Pan, S.J. and Yang, Q., 2010. A survey on transfer learning. IEEE Transactions on knowledge and data engineering, 22(10), pp.1345–1359.