Mask Classifier

CSE 5526 Final Project

Jude Rajasekera, Xinyue Li, and Zilin Wang

Project Idea

- Create a system that could detect faces in images, then classify them as masked or unmasked
- Create an application that could take in a video feed and overlay the labels in real time

Research

- *SSDMNV2: A real time DNN-based face mask detection system using single shot multibox detector and MobileNetV2*
  - Uses OpenCV's pre-trained facial recognition network to detect faces in images
  - Utilizes pre-trained MobileNetV2 as a feature extractor for mask classification using transfer learning

System Workflow

Pre-Processing

OpenCV Face Detector

(84, 51, 102, 78)

Pre-Processing

Mask Classifier

- MobileNetV2
- Our own CNN
- RBF Kernel SVM

"Mask"

Dataset

- Dataset from the SSDMNV2 paper.
  - 5,521 images with masks and 5,521 images without masks
  - Variety of lighting conditions, angles, resolutions, ethnicities, and genders
  - Some images have masks artificially drawn on faces

Image Preprocessing and Augmentation

Original

ColorJitter

RandomRotation

RandomAffine

RandomPerspective

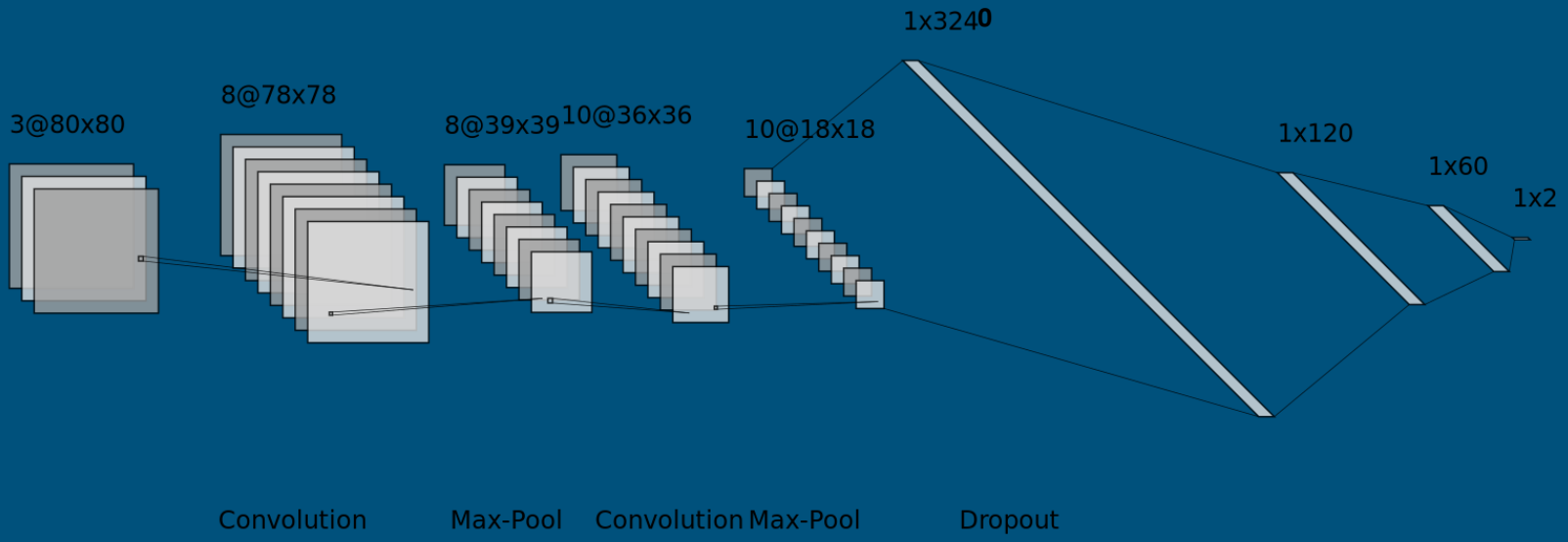
Normalize

SSDMNV2 Demo

Our Network Architecture

- Input images of 3 channels
- Convolutional Layer 1
  - 8 output channels
  - Kernel size of (3, 3)
- Convolutional Layer 2
  - 10 output channels
  - Kernel size of (4, 4)
- Each conv layer is followed by ReLU and Max-Pooling layer
- Max-Pooling Layer
  - Kernel size and stride of (2, 2)
- Dropout Layer
  - Dropout rate of 0.5
- Fully Connected Layers
  - 2 hidden layers
- Outputs an array of shape 1x2

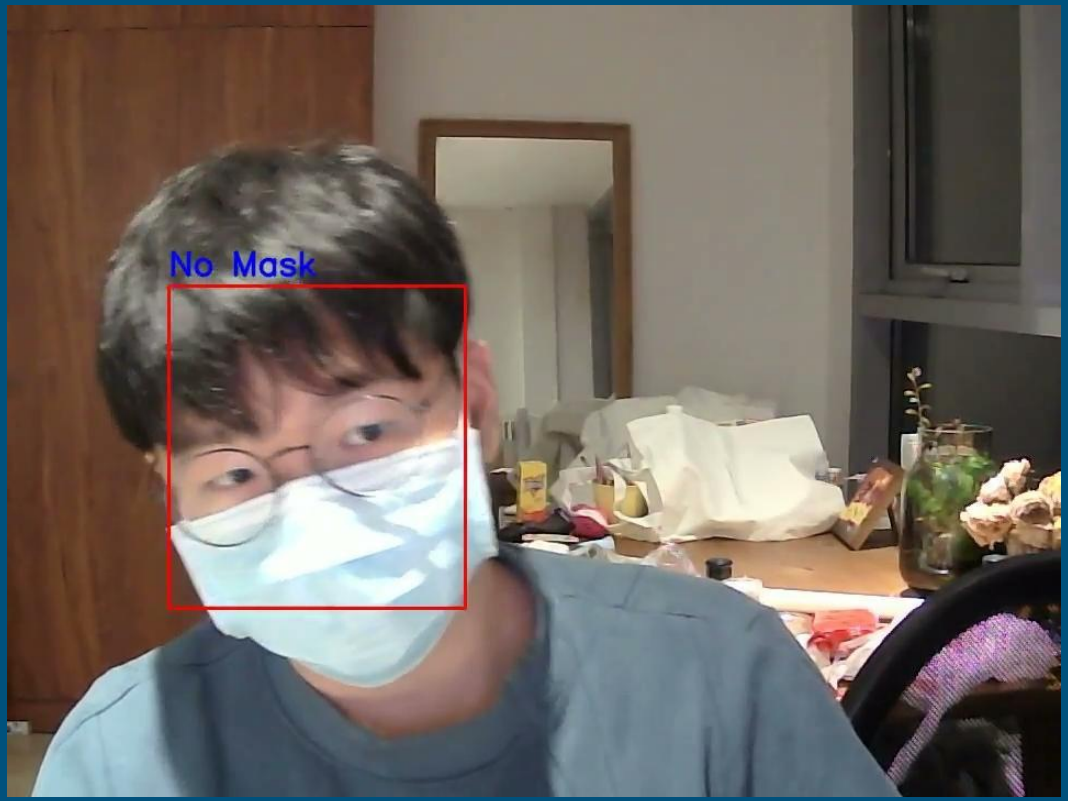
Our Network Architecture



Results

Model	Train Accuracy	Validation Accuracy
RBF Kernel SVM	0.9679	0.9586
MobileNetV2	0.9664	0.9782
CNN on Binary Classification	0.9949	0.9738

Demo without Image Augmentation or Dropout



Demo with Image Augmentation and Dropout

Adding a Third Class

- “No Mask”, “Mask”, “Incorrectly Wearing Mask”
- Combined Dataset from Kaggle
  - 853 images belonging to the 3 classes
  - Bounding boxes in the PASCAL VOC format
- Class Imbalance Problem
  - 4990, 6412, 99 images respectively
  - Solution: WeightedRandomSampler with replacement

Results

Model	Train Accuracy	Validation Accuracy
RBF Kernel SVM	0.9679	0.9586
MobileNetV2	0.9664	0.9782
CNN on Binary Classification	0.9949	0.9738
CNN on Three-Class Classification	0.9955	0.9746
CNN on Three-Class Classification with sampling	0.9551	0.8516

Conclusions

- The best performing network was our own CNN with Image Augmentation + Dropout
- Image augmentation greatly improves a network's ability to generalize to images not in the dataset
- Smaller filters in the CNN were more efficient and yielded higher classification accuracy for our problem
- Popular pre-trained models like MobileNetV2 are not always a better solution to a problem
- It is difficult to get good performance on a network without enough data

Future Work

- Try different approaches to deal with class imbalance problem for third label
  - Incorporate penalty terms
  - Create our own dataset with more images
- Try other learning architectures
  - Decision tree
  - Clustering Techniques

Thank you