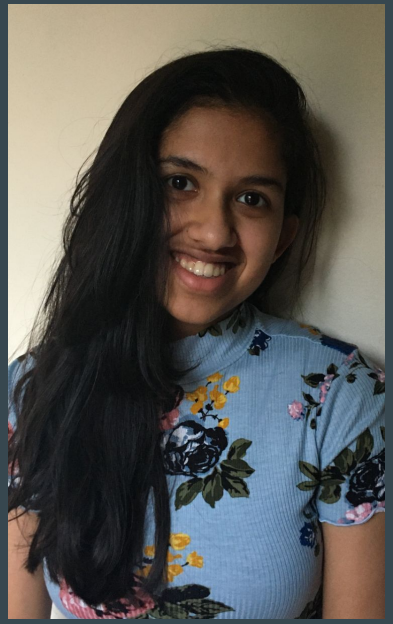
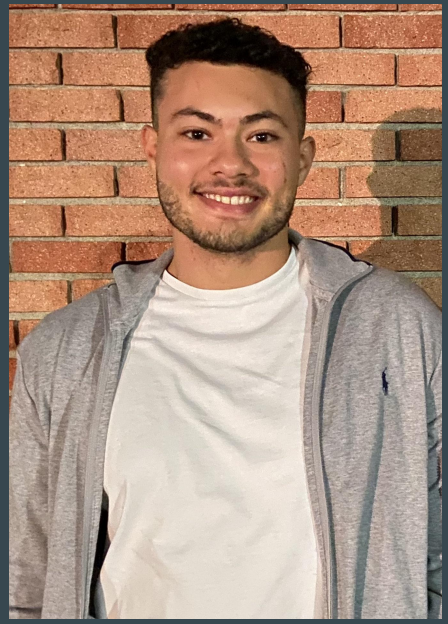
Real-Time Machine Learning

...

Facial and Mask Recognition in Real-Time



The Team



Nelusha Dias

Rutgers School of Engineering

ECE 2024

David Banyamin

Rutgers School of Engineering

ECE 2023

Overview

- Use Machine Learning for Real Time Tasks
  - Facial Recognition
    - ID Face in frame
    - ID Mask
  - Applications
    - Building security camera
    - Photo ID

Timeline

- Learning
  - ML
  - Python
  - Pytorch
  - Hardware
- Algorithms
  - Facial Recognition
  - Put names to faces
  - Saving unknown & masked faces
  - Check for mask using CNN

Program Overview

- Combined useful aspects of two codes
  - Doorcam
  - Facial Recognition
- Known Face Identification
- Saves Unknown Faces
- Mask Identification

```
# Main Loop
def main_loop():
        unknown_number = 1
        mask_number = 1
        while True:
                # Grab a single frame of video
                ret, frame = video_capture.read()

                # Resize frame of video to 1/4 size for faster face recognition processing
                small_frame = cv2.resize(frame, (0, 0), fx=0.25, fy=0.25)

                # Convert the image from BGR color (which OpenCV uses) to RGB color (which face_recognition uses)
                rgb_small_frame = small_frame[:, :, ::-1]


                # Find all the face locations and face encodings in the current frame of the video
                face_locations = face_recognition.face_locations(rgb_small_frame)
                face_encodings = face_recognition.face_encodings(rgb_small_frame, face_locations)

                # Loop through each detected face and see if it is one we have seen before
                # If so, we'll give it a label that we'll draw on top of the video
                if process_this_frame:
                        mask = False

                # Find all the faces and face encodings in the current frame of video
                        face_locations = face_recognition.face_locations(rgb_small_frame)
                        face_encodings = face_recognition.face_encodings(rgb_small_frame, face_locations)
                        face_names = []
                        for face_encoding in face_encodings:


                        # See if the face is a match for the known face(s)
                                matches = face_recognition.compare_faces(known_face_encodings, face_encoding)
                                name = "Unknown"



                                # If matches use the known face with the smallest distance to the new face
                                face_distances = face_recognition.face_distance(known_face_encodings, face_encoding)
                                best_match_index = np.argmin(face_distances)
                                if matches[best_match_index] and mask == False:
                                        name = known_face_names[best_match_index]

                                if name == "Unknown" and mask == False:
                                        image = frame
                                        unknown_number_string = str(unknown_number)
                                        image_name = "unknown" + unknown_number_string + ".jpg"
```

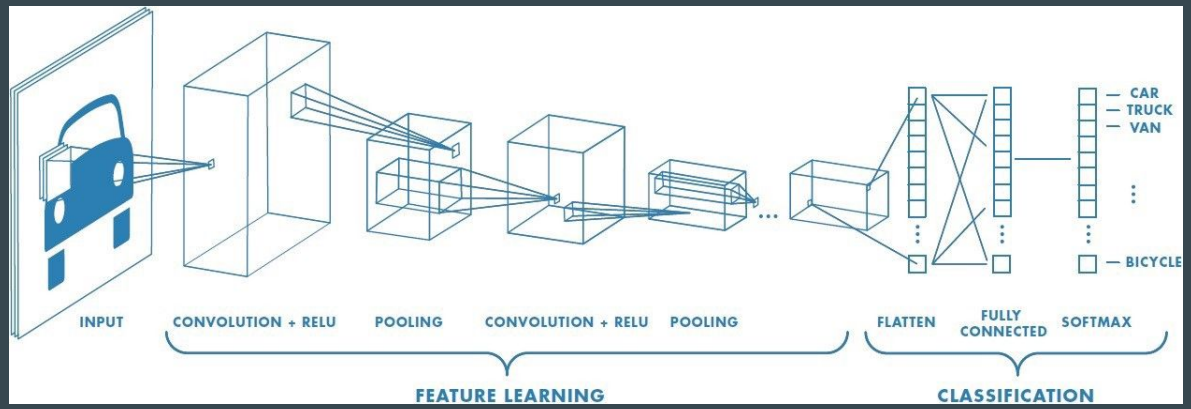
Convolutional Neural Network (CNN)

General Knowledge

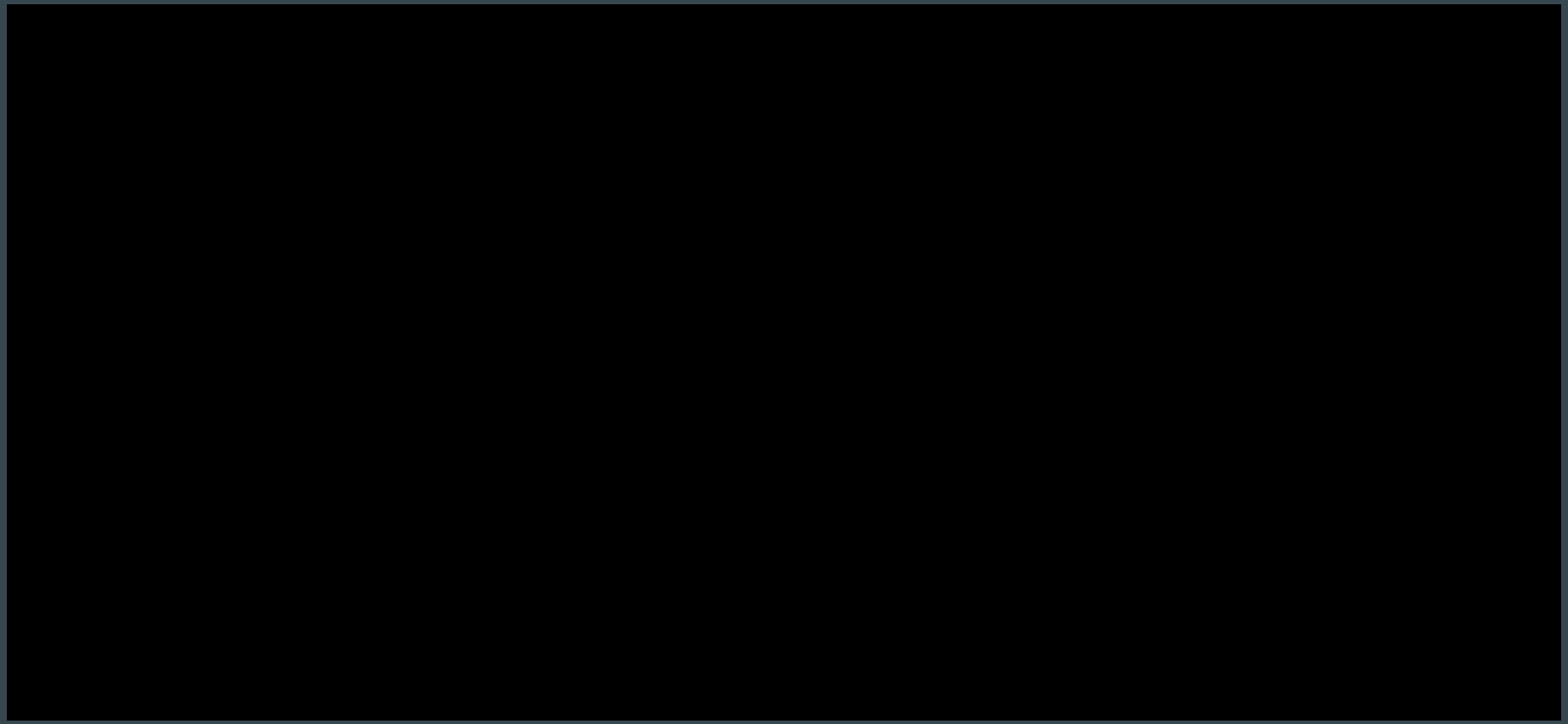
- Algorithm takes input image and identifies what it is
- Image goes through layers which apply filters onto the image

Our Program

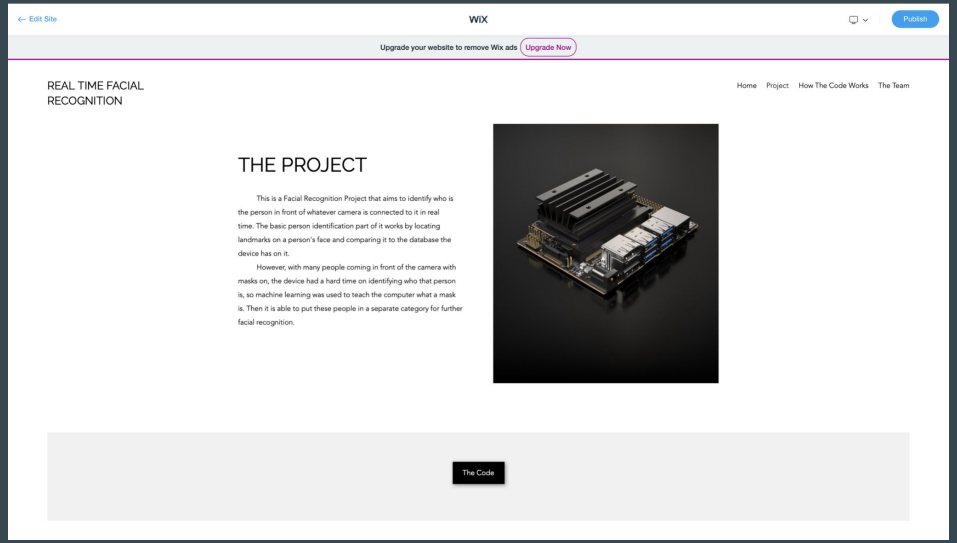
- Two 2D convolutional layers
- Two pooling layers
- Three fully-connected layers
- Learning rate 0.001
- 4 epochs and batch size of 4



Video Demonstration

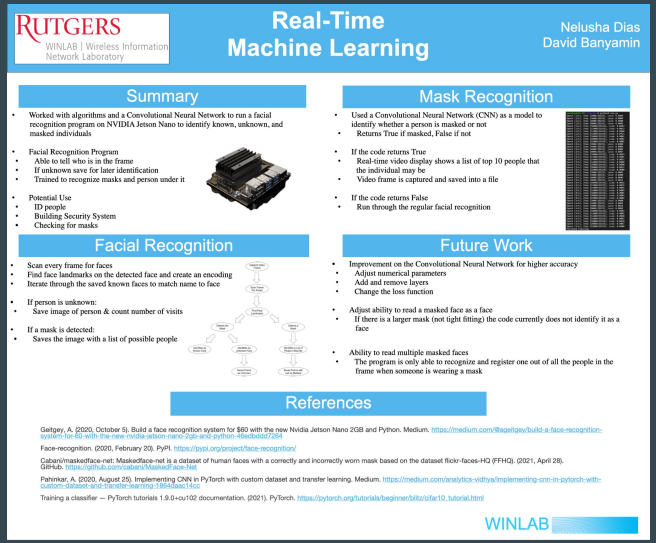


Website



https://realtimeml1.wixsite.com/facial-recognition

Poster



Conclusion & Future

Results:

- Got a functioning program
  - Identify the person in front of the camera
  - Saves unknown and masked individuals

```
[root@node21-32:~/project# python3 cnn.py
Epoch [1/4], Step [2000/23223], Loss: 0.0003
Epoch [1/4], Step [4000/23223], Loss: 0.0003
Epoch [1/4], Step [6000/23223], Loss: 0.0000
Epoch [1/4], Step [8000/23223], Loss: 0.0002
Epoch [1/4], Step [10000/23223], Loss: 0.0032
Epoch [1/4], Step [12000/23223], Loss: 0.0012
Epoch [1/4], Step [14000/23223], Loss: 0.0000
Epoch [1/4], Step [16000/23223], Loss: 0.0914
Epoch [1/4], Step [18000/23223], Loss: 0.0001
Epoch [1/4], Step [20000/23223], Loss: 0.0002
Epoch [1/4], Step [22000/23223], Loss: 0.0001
Epoch [2/4], Step [2000/23223], Loss: 0.0000
Epoch [2/4], Step [4000/23223], Loss: 0.0006
Epoch [2/4], Step [6000/23223], Loss: 0.0002
Epoch [2/4], Step [8000/23223], Loss: 0.0007
Epoch [2/4], Step [10000/23223], Loss: 0.0033
Epoch [2/4], Step [12000/23223], Loss: 0.0001
Epoch [2/4], Step [14000/23223], Loss: 0.0021
Epoch [2/4], Step [16000/23223], Loss: 0.0000
Epoch [2/4], Step [18000/23223], Loss: 0.0000
Epoch [2/4], Step [20000/23223], Loss: 0.0012
Epoch [2/4], Step [22000/23223], Loss: 0.0000
Epoch [3/4], Step [2000/23223], Loss: 0.0001
Epoch [3/4], Step [4000/23223], Loss: 0.0000
Epoch [3/4], Step [6000/23223], Loss: 0.0002
Epoch [3/4], Step [8000/23223], Loss: 0.0011
Epoch [3/4], Step [10000/23223], Loss: 0.0016
Epoch [3/4], Step [12000/23223], Loss: 0.0015
Epoch [3/4], Step [14000/23223], Loss: 0.0004
Epoch [3/4], Step [16000/23223], Loss: 0.0003
Epoch [3/4], Step [18000/23223], Loss: 0.0001
Epoch [3/4], Step [20000/23223], Loss: 0.0004
Epoch [3/4], Step [22000/23223], Loss: 0.0002
Epoch [4/4], Step [2000/23223], Loss: 0.0000
Epoch [4/4], Step [4000/23223], Loss: 0.0002
Epoch [4/4], Step [6000/23223], Loss: 0.0012
Epoch [4/4], Step [8000/23223], Loss: 0.0002
Epoch [4/4], Step [10000/23223], Loss: 0.0002
Epoch [4/4], Step [12000/23223], Loss: 0.0002
Epoch [4/4], Step [14000/23223], Loss: 0.0000
Epoch [4/4], Step [16000/23223], Loss: 0.0002
Epoch [4/4], Step [18000/23223], Loss: 0.0010
Epoch [4/4], Step [20000/23223], Loss: 0.0007
Epoch [4/4], Step [22000/23223], Loss: 0.0000
Finished Training
```

Improvement:

- CNN
- Adjust ability to read a masked face
- Ability to read multiple masked faces

Potential Use:

- Security camera
- Ensure people are wearing masks before entrance

References

Geitgey, A. (2020, October 5). Build a face recognition system for $60 with the new Nvidia Jetson Nano 2GB and Python. Medium. https://medium.com/@ageitgey/build-a-face-recognition-system-for-60-with-the-new-nvidia-jetson-nano-2gb-and-python-46edbddd7264

Face-recognition. (2020, February 20). PyPI. https://pypi.org/project/face-recognition/

Cabani/maskedface-net: Maskedface-net is a dataset of human faces with a correctly and incorrectly worn mask based on the dataset flickr-faces-HQ (FFHQ). (2021, April 28). GitHub. https://github.com/cabani/MaskedFace-Net

Pahinkar, A. (2020, August 25). Implementing CNN in PyTorch with custom dataset and transfer learning. Medium. https://medium.com/analytics-vidhya/implementing-cnn-in-pytorch-with-custom-dataset-and-transfer-learning-1864daac14cc

Training a classifier — PyTorch tutorials 1.9.0+cu102 documentation. (2021). PyTorch. https://pytorch.org/tutorials/beginner/blitz/cifar10_tutorial.html

Thank You!
Any Questions?