

# CHARACTER CLASSIFICATION IN VIDEOS

CS 131 Final Project | Fall 2022

Created by Alexis Echano

Parks and  
Recreation



# BACKGROUND

- Fun way to combine computer vision and a popular sitcom with an ensemble cast
- Efficiency when editing, applying CGI, and even labeling
- No one has really done this!

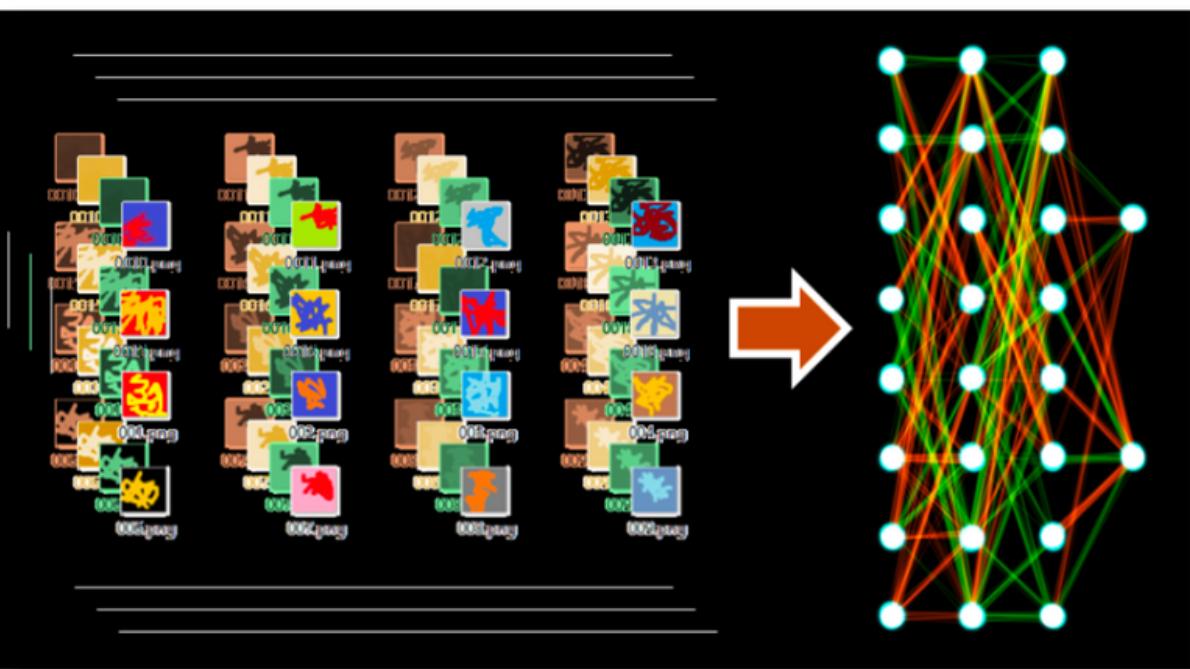


# IMPLEMENTATION

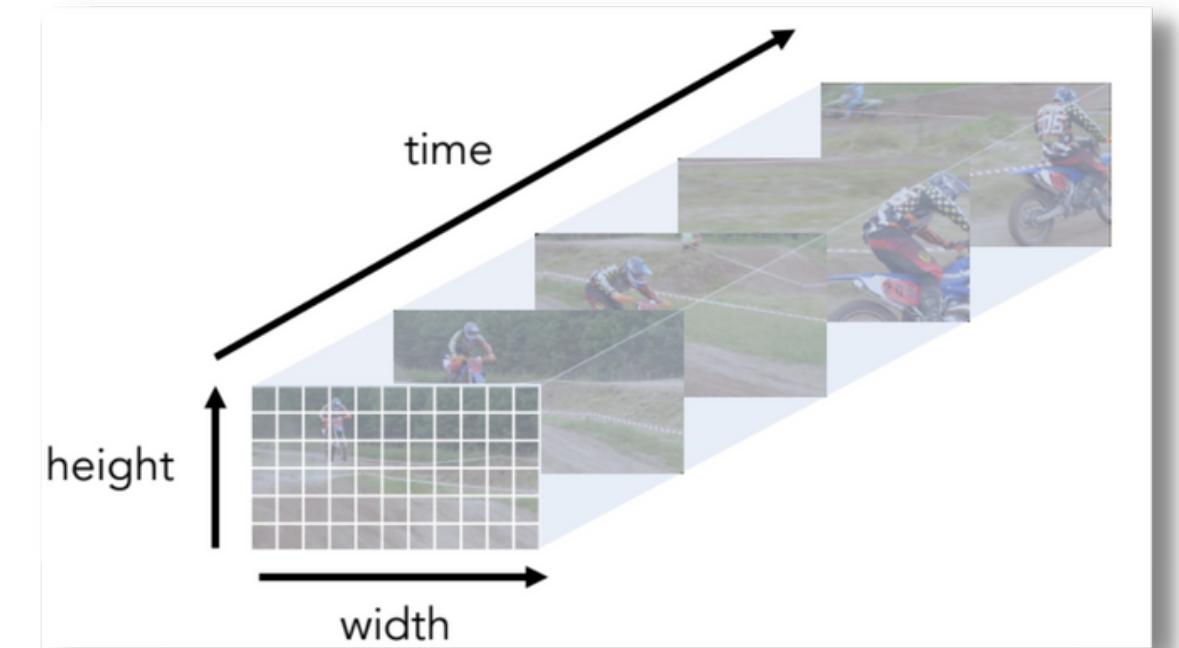
Breaking down the steps...



Collecting data



Training a model



Understanding videos

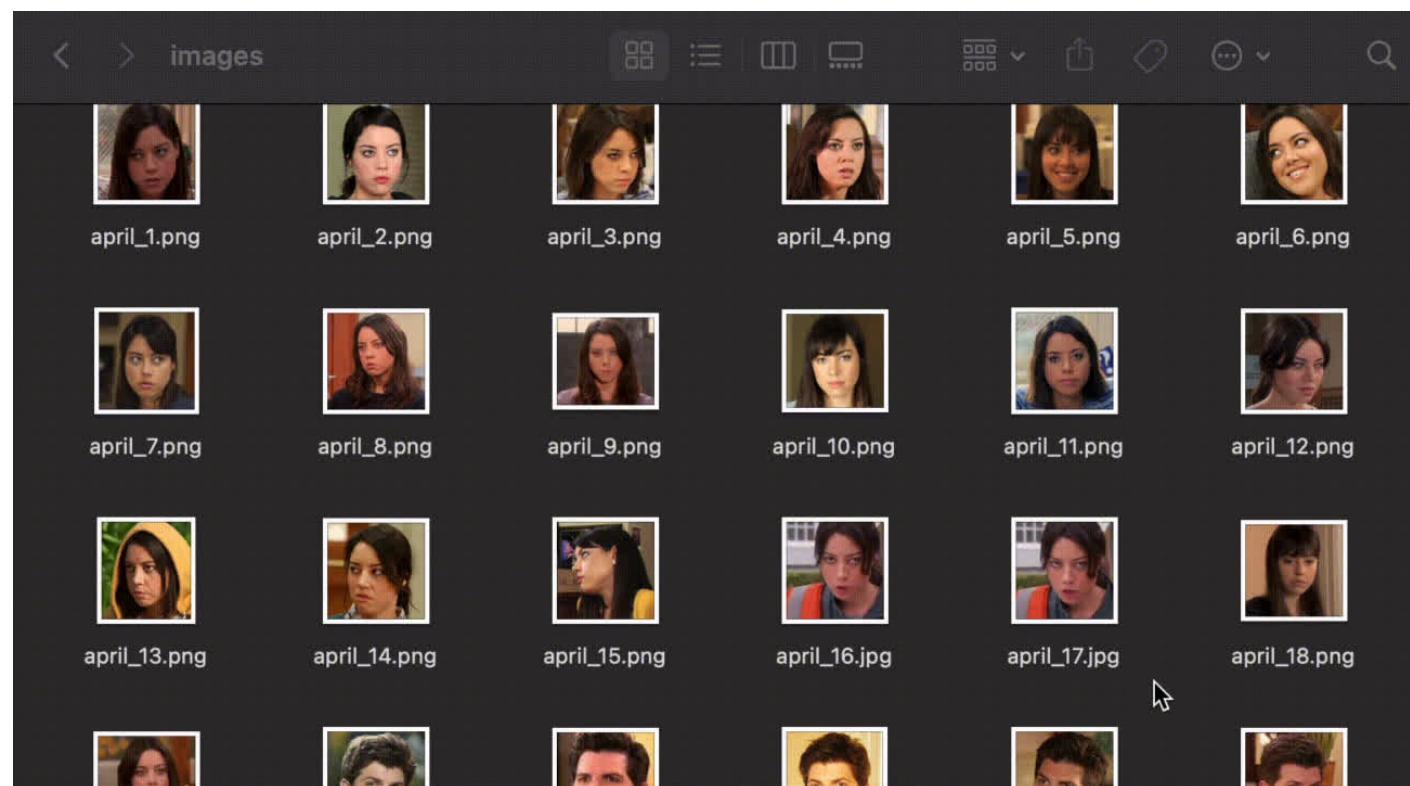
# COLLECTING DATA

Labeling the data and creating the dataset

- Had to manually collect over a hundred 128x128 images of characters
- Created a custom PyTorch dataset called ParksAndRecDataset that read in a CSV of hand-labeled data

```
class ParksAndRecDataset(Dataset):
    """Parks and Recreation character images dataset."""

    def __init__(self, csv_file, root_dir, transform=None):
        """
        Args:
            csv_file (string): Path to the csv file with annotations.
            root_dir (string): Directory with all the images.
            transform (callable, optional): Optional transform
                on a sample.
        """
        self.read_file = pd.read_csv(csv_file)
        self.root_dir = root_dir
        self.transform = transform
```



# TRAINING & TESTING

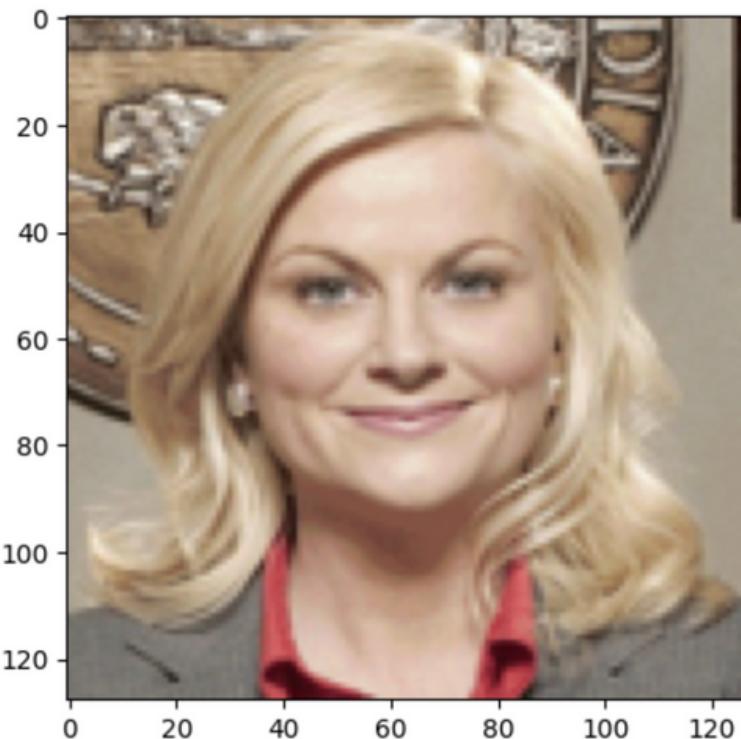
Training the model on character images

~71%

Test Accuracy

- Built upon network from Homework 7
  - Added extra layers
  - Incorporated batch norm and dropout
  - More improvements!

```
Accuracy for character leslie is: 100.0 %
Accuracy for character ben    is: 100.0 %
Accuracy for character ron   is: 33.3 %
Accuracy for character donna is: 80.0 %
Accuracy for character april is: 75.0 %
```



real: leslie  
predicted: leslie

```
# set up 2d convolutions
self.conv_1 = nn.Conv2d(3, 64, self.kernel_size, 1, self.padding)
self.norm_1 = nn.BatchNorm2d(64)
self.pool_1 = nn.MaxPool2d(2, 2)

self.conv_2 = nn.Conv2d(64, 128, self.kernel_size, 1, self.padding)
self.norm_2 = nn.BatchNorm2d(128)
self.pool_2 = nn.MaxPool2d(2, 2)

self.conv_3 = nn.Conv2d(128, 128, self.kernel_size, 1, self.padding)
self.norm_3 = nn.BatchNorm2d(128)
self.pool_3 = nn.MaxPool2d(2, 2)

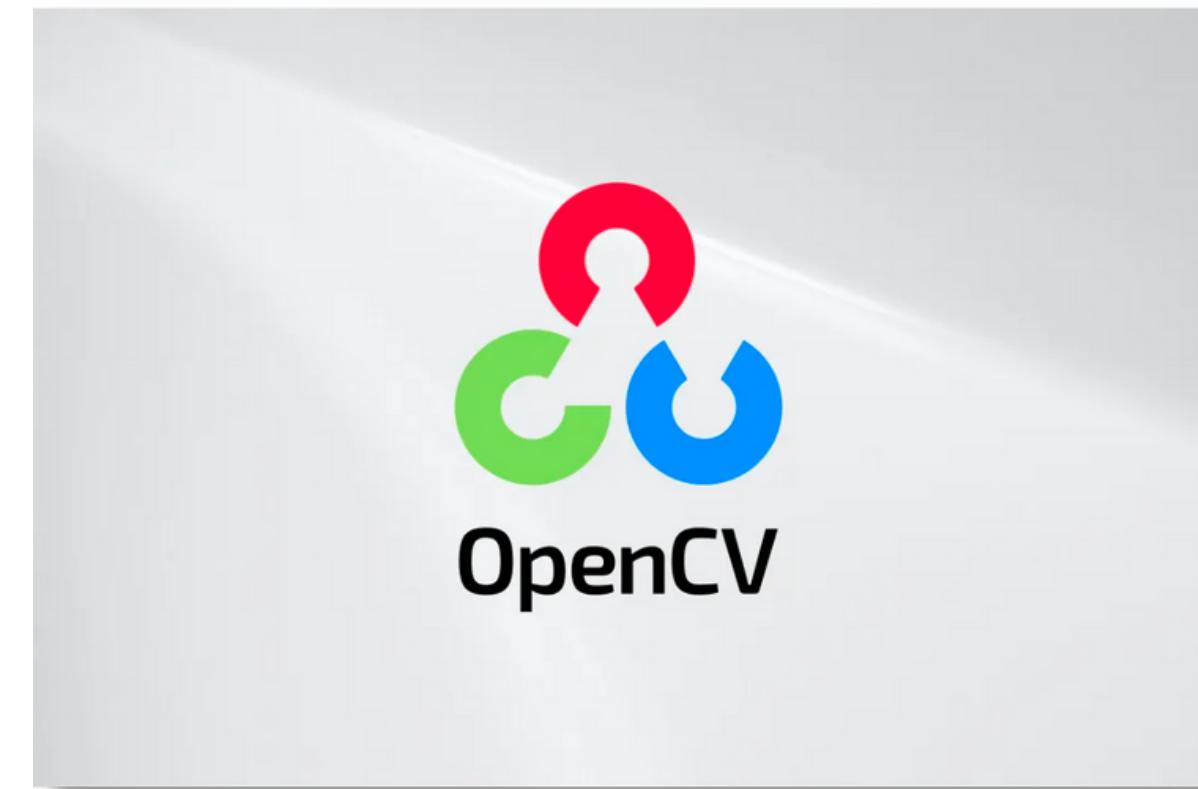
self.conv_4 = nn.Conv2d(128, 128, self.kernel_size, 1, self.padding)
self.norm_4 = nn.BatchNorm2d(128)
self.pool_4 = nn.MaxPool2d(2, 2)

# set up fully connected layer
self.num_lin_inputs = 512 * 4 * 4
self.fc_1 = nn.Linear(self.num_lin_inputs, 5)

# add dropout
self.dropout = nn.Dropout(0.4)
```

# VIDEO CLIPS

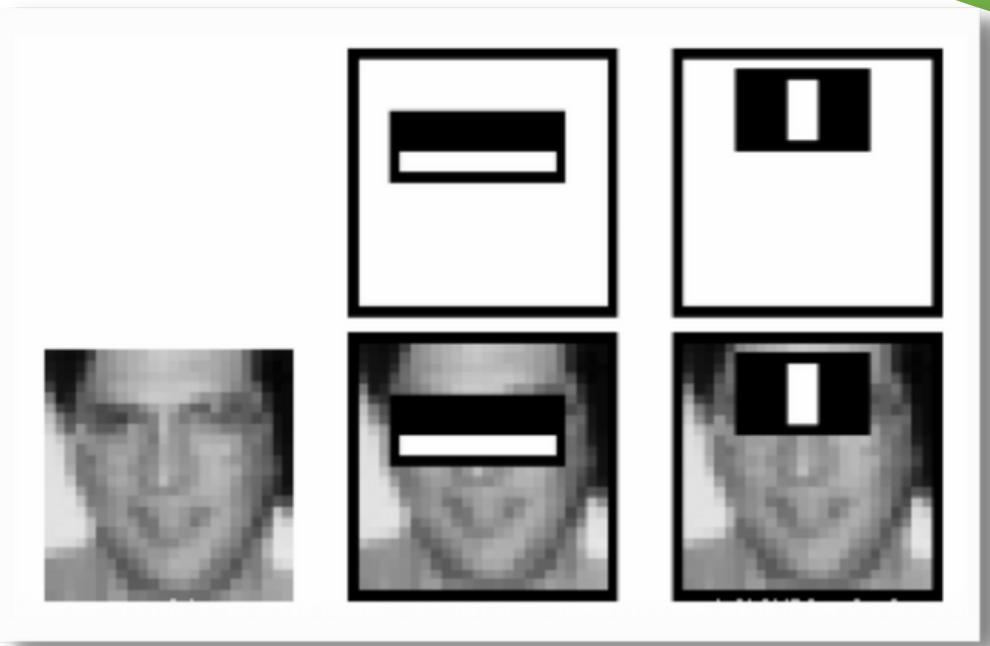
- Lots of learning happened here!
- Topics from Homework 5 + OpenCV
  - Broke down clips into image frames and reduced FPS
  - Animated frames to recreate video
  - Made video content digestible for model



```
while success:  
    if (frame_count % target_div) == 0:  
        cv2.imwrite(f"./gen_frames/{class_name}/frame{file_count}.jpg", image)  
        success, image = video.read()  
        file_count += 1  
        frame_count += 1  
    else:  
        success = video.grab()  
        frame_count += 1  
video.release()
```

# FACIAL RECOGNITION

- Used bounding boxes, though more complex than HW5...
  - OpenCV's pre-trained Cascade Classifier
  - HAAR Cascade - only considering contours of the face important
- After finding the face and drawing a box, add padding and get final input image!

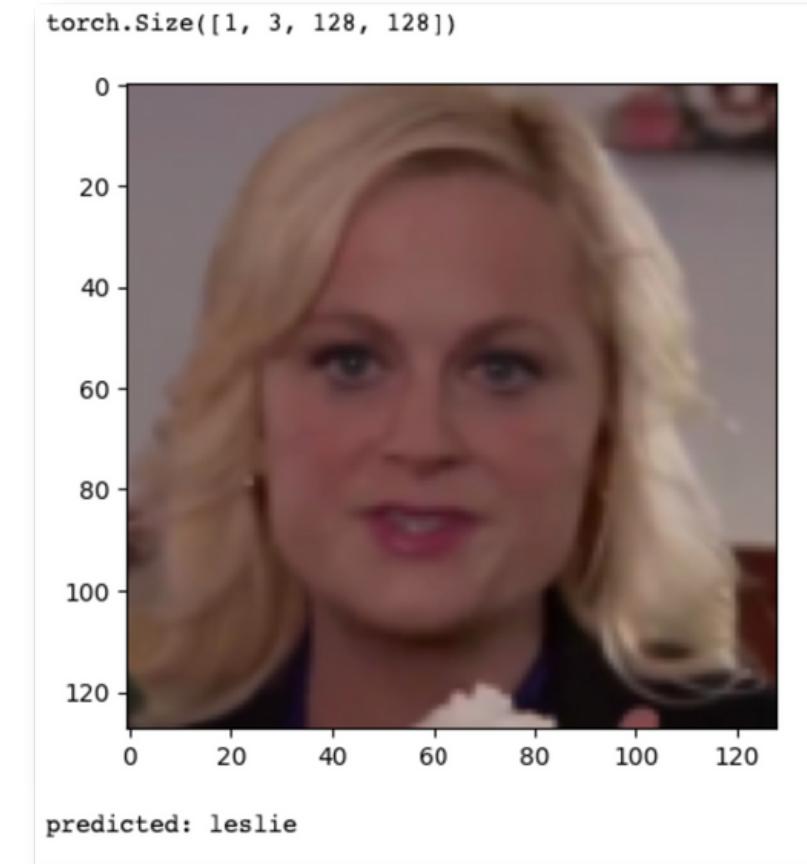


# RESULTS

- Using an assortment of clips from an episode of *Parks and Recreation*
- Calculated the mode of all of the frames' predicted classification
- Achieved about the same accuracy as image-only model testing
- Interesting results as features like gender and hair color play a role in the mismatches

71.4%

Test Accuracy



File Name	Actual Character	Predicted	Correct?
1.mov	leslie	leslie	yes
2.mov	donna	donna	yes
3.mov	april	april	yes
4.mov	ron	april	no
5.mov	leslie	april	no
6.mov	ben	ron	no
7.mov	april	april	yes
8.mov	ron	ron	yes
9.mov	leslie	leslie	yes
10.mov	ben	ben	yes
11.mov	leslie	april	no
12.mov	april	april	yes
13.mov	ben	ben	yes
14.mov	donna	donna	yes

# ANY QUESTIONS?

CHECK OUT THE CODE HERE:  
[shorturl.at/qrE28](https://shorturl.at/qrE28)

\*will probably make repo private later on\*