DTSA5509 Supervised Learning

Final Project: Sunglasses Image Classifier

Matt Schurmann
masc3796@colorado.edu
Nov 23 2023

Data Set and Problem Description (1)

- Data Set: Mitchell,Tom. (1999). CMU Face Images. UCI Machine Learning Repository. https://doi.org/10.24432/C5JC79.
- Type of Learning: Supervised Learning
- Type of Problem: Image Classification

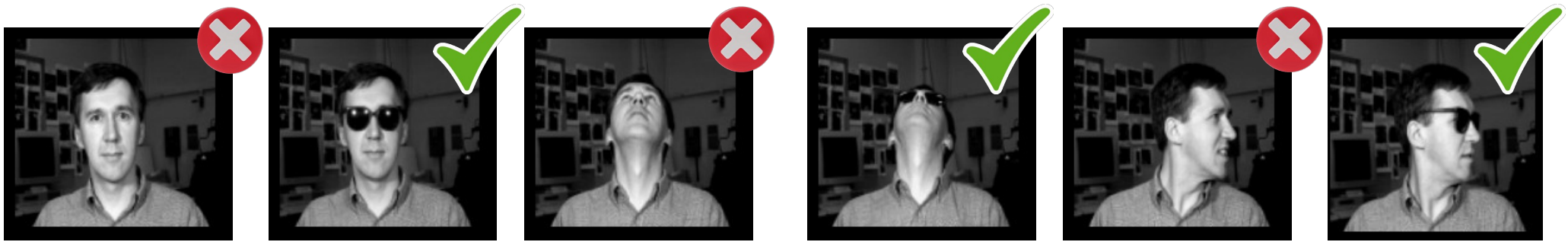
Python Libraries Used:

Scikit-learn
matplotlib
seaborn
pandas
numpy

Data Set and Problem Description (2)

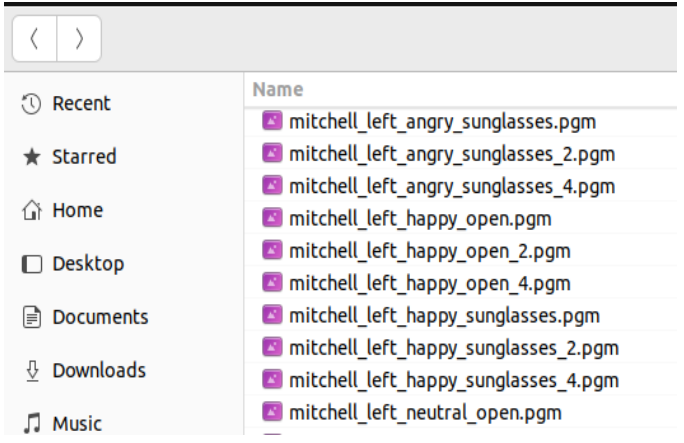
Number of Images: 640
File Format: Portable Greyscale Map (.pgm)
Color Palette: Greyscale
Dimensions: 3 copies of each image, (120x128), (60x64), (30x32)
Modalities:

Unique Faces – 20 different people
Direction - Left, Right, Up, Straight
Mood - Happy, Sad, Angry, Neutral
Sunglasses - On, Off



Data Cleaning and Preparation (1)

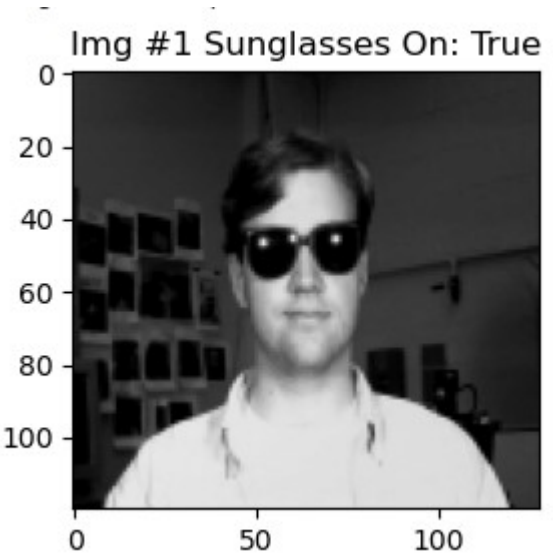
- Training data labels are encoded in the file names of each image and not in any tabular format.
- Mis-captured images need to be removed.
- Each image is duplicated 3 times in different resolutions. I have selected only the highest resolution versions for this project.
- Some label classes are not binary (e.g. more than 2 labels). For example, "mood" can take values (happy, sad, angry, neutral). These will be split into binary labels (mood_happy, mood_sad, mood_angry, mood_neutral) which have values (1 or 0).



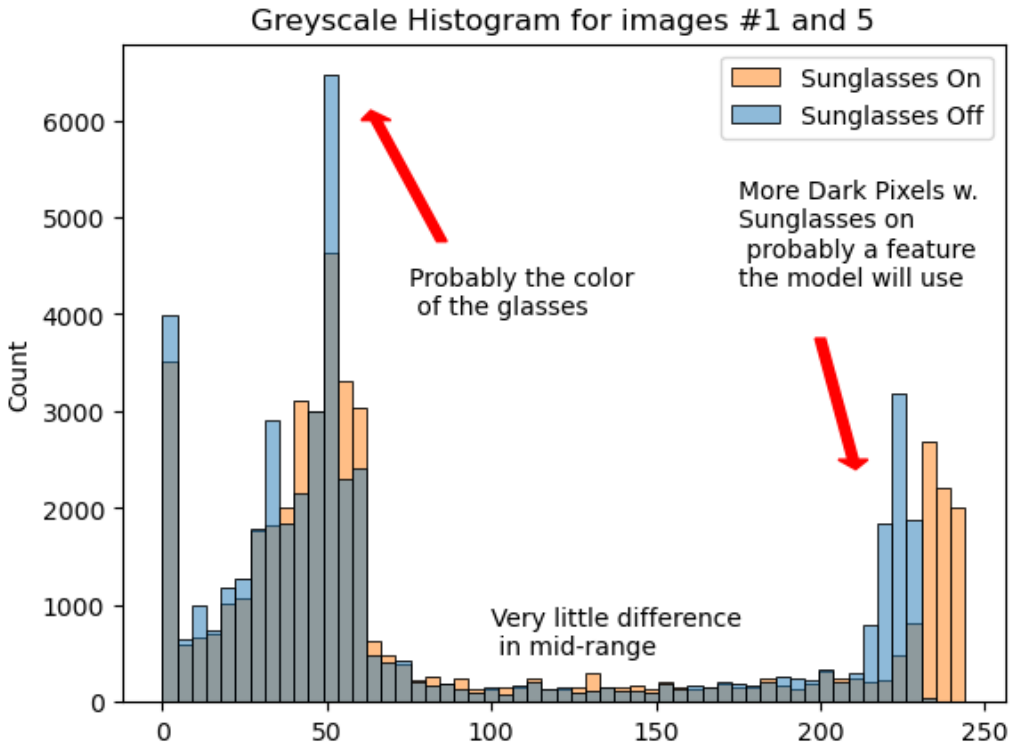
Data Cleaning and Preparation (2)

```
#first un-pickle the data and get ready to rock-and-roll
with open("faces.pickle", "rb") as f:
    df = pickle.load(f)
```

```
#Select a random image from the data set and inspect the format of
X = df["pixels"]
y = df["sunglasses_on"]

def show_image(X, y, index, title=""):
    fig = plt.figure(figsize=(3,3))
    plt.imshow(X[index].reshape(120, 128, 3), cmap='gray')
    plt.title(title + "Sunglasses On: " + str(bool(y[index])))
    plt.show()

print("Pixel Data information...")
print("image data dtype", X[0].dtype)
print("image data shape", X[0].shape)

#select two similar images one with sunglasses on/off
show_image(X, y, 1, "Img #1 ")
```



Exploratory Data Analysis (EDA)



- Features are not easily separable for image data .
- Here we attempt to look for possible differences based on luminosity.
- Sunglasses are dark…

Modeling and Training (1)

Multiple Models Evaluated

- K-Nearest Neighbors (KNN)
- Support Vector Machine (SVM)
- Gaussian Naive Bayes (NB) – not covered in the course
- Random Forest Ensemble (RF)

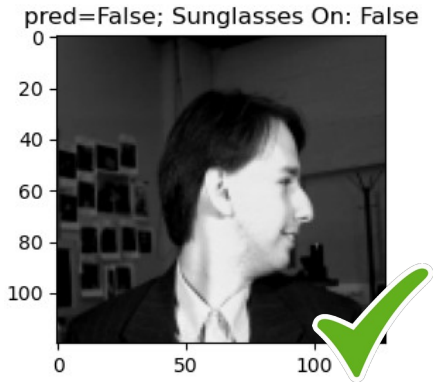
Methods to improve robustness

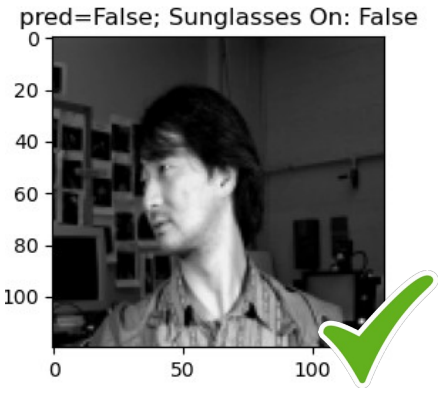
- Manual Tuning to determine parameter ranges
- Test/train split of 50% to ensure truly unseen data is tested
- Automated Tuning using Grid-Search, with K-Folds Cross-Validation
- Scoring using Accuracy (data set is balanced)
- Confusion Matrix to identify FP and FN rates if asymmetrical

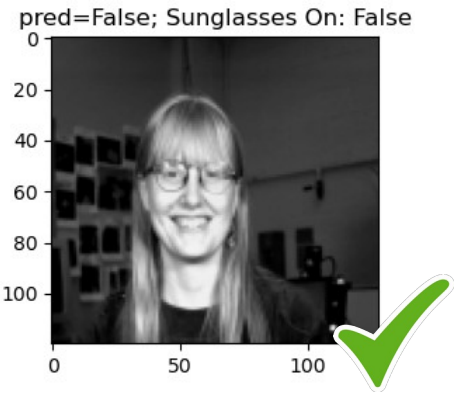
Modeling and Training (2)

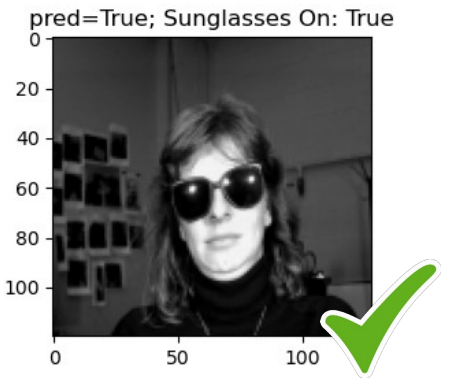
Rank	Model	Test Accuracy	Training Time (s)	Test Time (s)
1	SVM	0.9231	1.4911	0.9798
2	Random Forest	0.8558	1.5423	0.0189
3	KNN	0.7436	0.0033	0.1073
4	Naive Bayes	0.6378	0.08925	0.1275

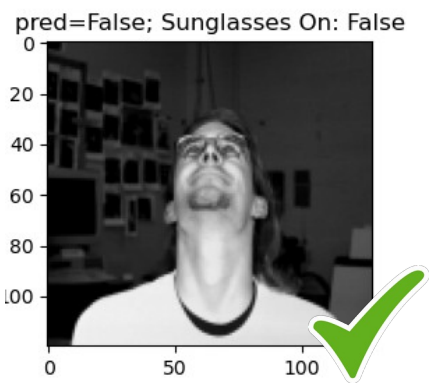
Results of SVM Model (Test Data)

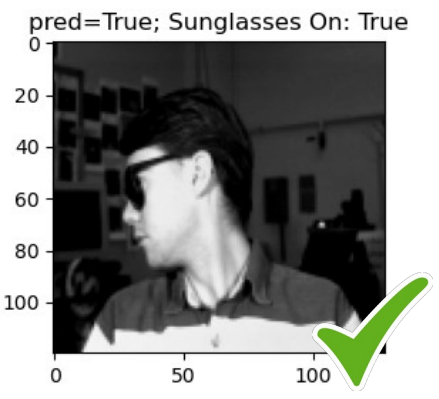


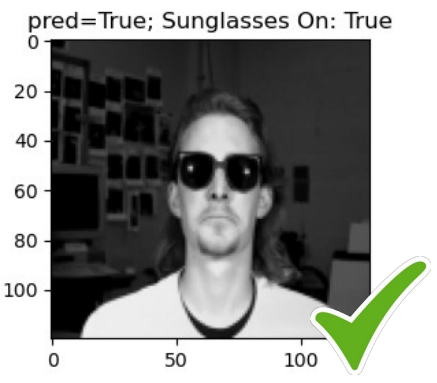


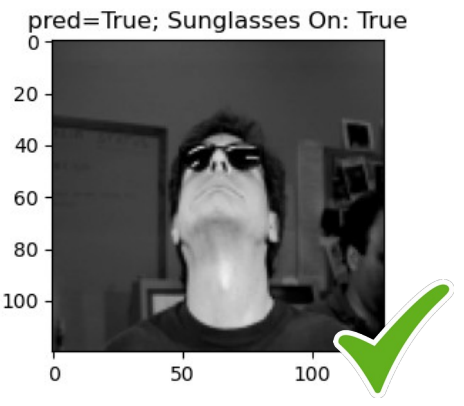










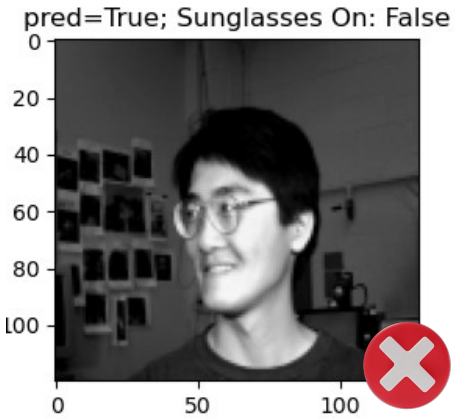


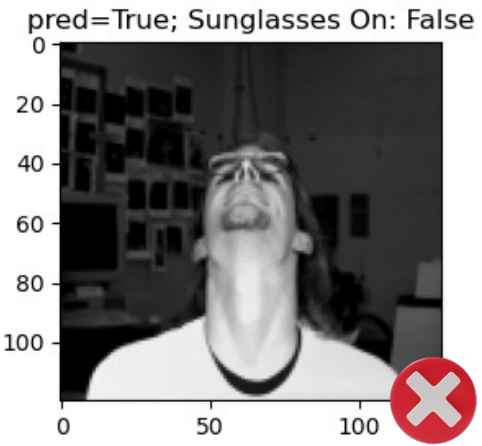
92% Test Accuracy overall

Examining False Positives/Negatives

False Positives

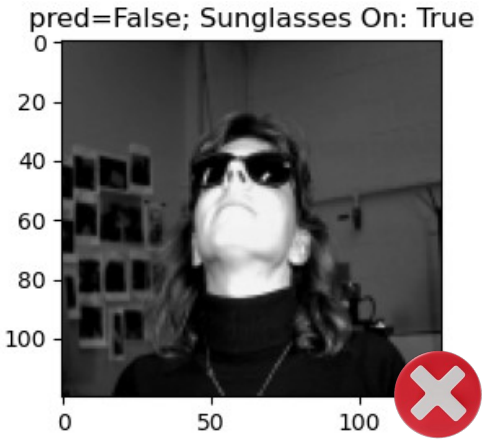
Several false-positives wearing glasses
(but not sunglasses)

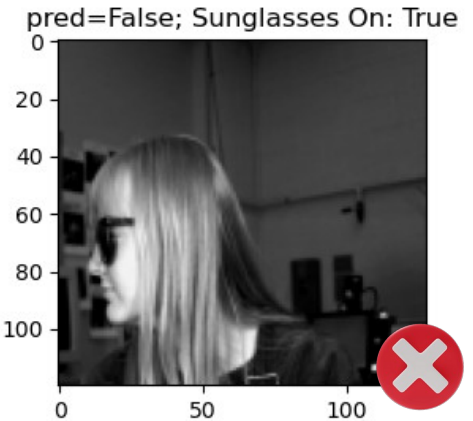




False Negatives

Several examples with reflections in
glasses, and non-forward head position





Conclusions and Discussion

Learning Outcomes

- Hands-on, real-World use of Sklearn classification models
- Challenges of image classification (high dimensionality data)
- Challenges of data labeling and feature identification

Suggestions for Improvement

- More efficient models than SVM, trade-off with accuracy if needed
- Use a larger training data set
- Investigate lower resolution images to improve speed
- Perform a more exhaustive parameter grid-search on a higher performance computer
- Include color photographs - presumably color would differentiate the sunglasses more vs. dark spots in grey-scale images.

Thank You!