


Real-time Distracted Driver Detection using Deep Learning

Ming Tang

COURSE CODE :- ECE_5831

Pattern Recognition and Neural Network

Background information

Function: This project aims to develop a real-time system that analyzes visual cues from in-vehicle cameras to accurately predict various driver distraction states, including texting, talking on the phone, or looking away.

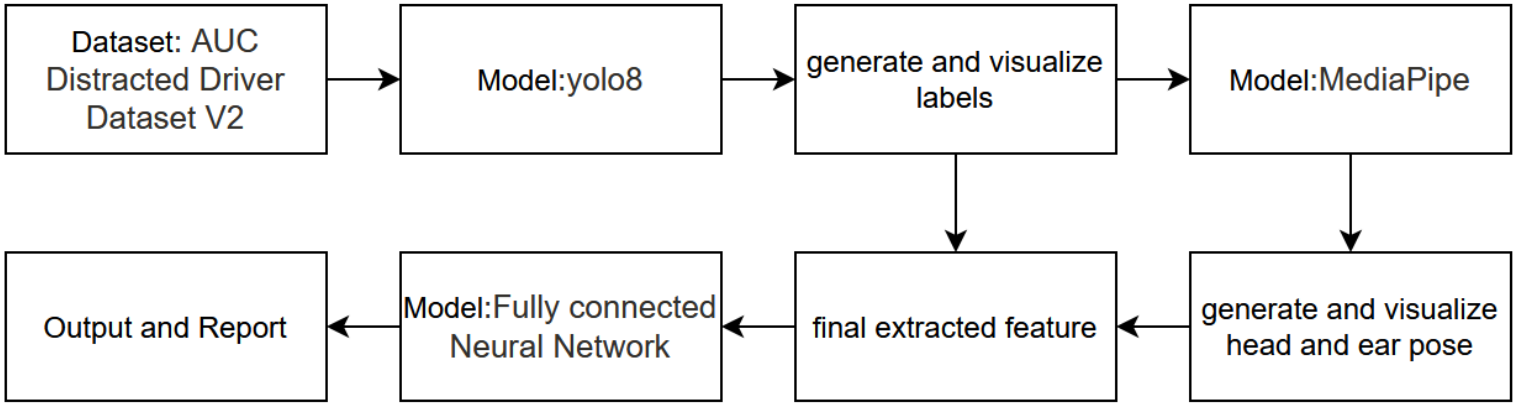
Benefit: The system's primary benefit is to enhance road safety by providing timely alerts for driver inattention, thereby minimizing accidents, reducing fatalities, and improving overall driving behavior.

Dataset: AUC Distracted Driver Dataset (V2)

This dataset comprises images from multiple camera angles (Camera 1 and Camera 2), categorized into ten distinct driver states (c0-c9), including safe driving and various forms of distraction. It is specifically designed for research in driver distraction identification and allows for evaluation based on driver splits.

- **c0:** Safe Driving
- **c1:** Text Right
- **c2:** Talk Right
- **c3:** Text Left
- **c4:** Talk Left
- **c5:** Adjust the Radio
- **c6:** Drink
- **c7:** Reaching Behind
- **c8:** Hair and Makeup
- **c9:** Talk to Passengers

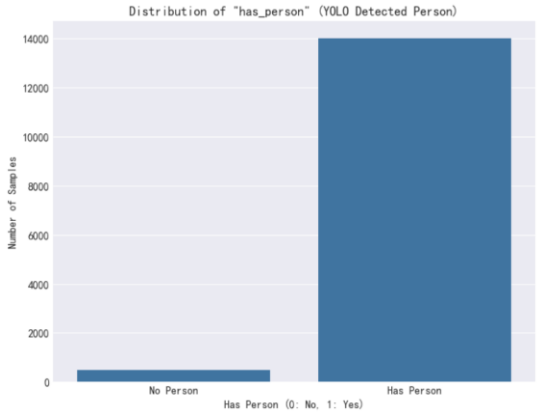
Flowchart


Dataset: AUC Distracted Driver Dataset V2
Model:yolo8
generate and visualize labels
Model:MediaPipe
generate and visualize head and ear pose
final extracted feature
Model:Fully connected Neural Network
Output and Report

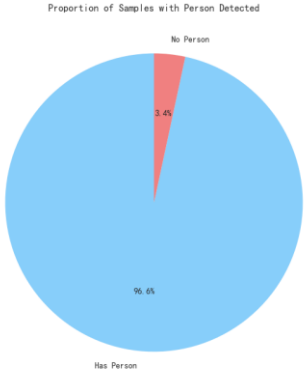
Overview of YOLO's Role

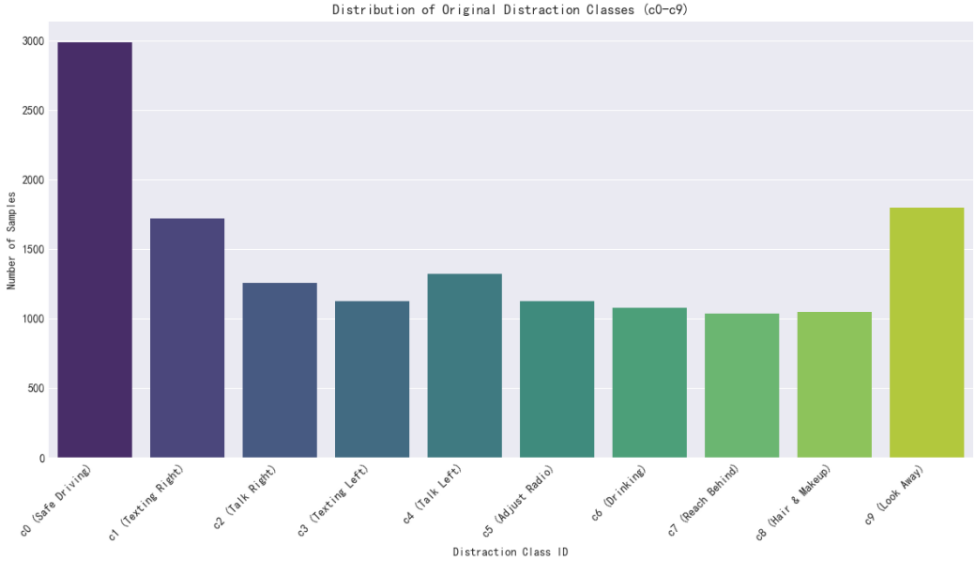
YOLOv8 serves as the initial computer vision module in our pipeline, primarily responsible for **object detection**. Its task is to identify key elements in the dataset images: the **driver (person)**, and any **distracting objects** like cell phones or bottles. The output of YOLOv8 forms the foundation of our pseudo-labels, which are crucial for subsequent feature extraction.

Yolo V8 Output Visualization



Probability of detecting a person





Data distribution

Yolo V8 Output Visualization

Person

Person cellphone

Person cellphone

Person cellphone

Person bottle

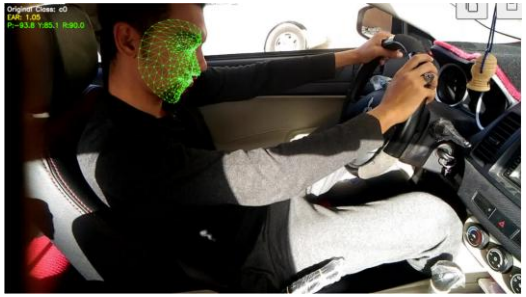
Yolo V8 summary

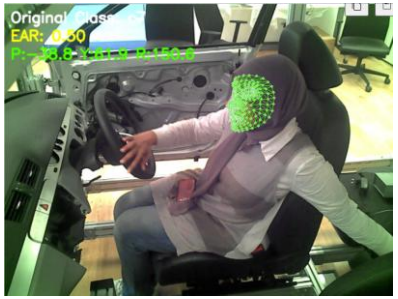
- **Strength:** YOLOv8 reliably detects the driver ('person’) with high accuracy but is not so reliable on primary distracting objects ('cellphone', 'bottle.
- **Foundation for Feature Extraction:** The precise bounding box for 'person' is vital, as it allows MediaPipe to focus its facial analysis on the correct region, enhancing the accuracy of EAR and head pose estimation

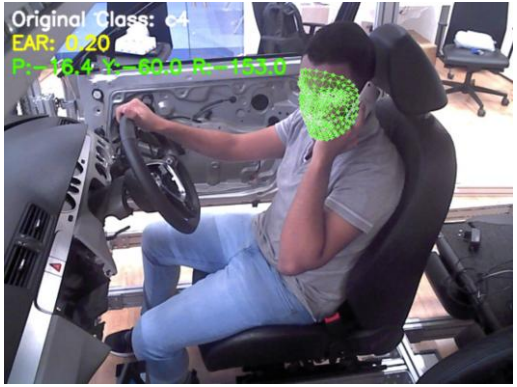
Use Mediapipe to extract feature from ‘person’

Following YOLOv8's object detection, our system utilizes **Google's MediaPipe Face Mesh**, a robust and efficient framework for **high-fidelity 3D facial landmark detection**. MediaPipe's role is critical for extracting more subtle and detailed visual cues about the driver's state. Specifically, it allows us to precisely calculate the **Eye Aspect Ratio (EAR)** to monitor for signs of inattention, and to estimate the **Head Pose (Pitch, Yaw, Roll angles)** to determine the driver's head orientation. These fine-grained facial features provide essential information for classifying various non-object-based distraction behaviors.

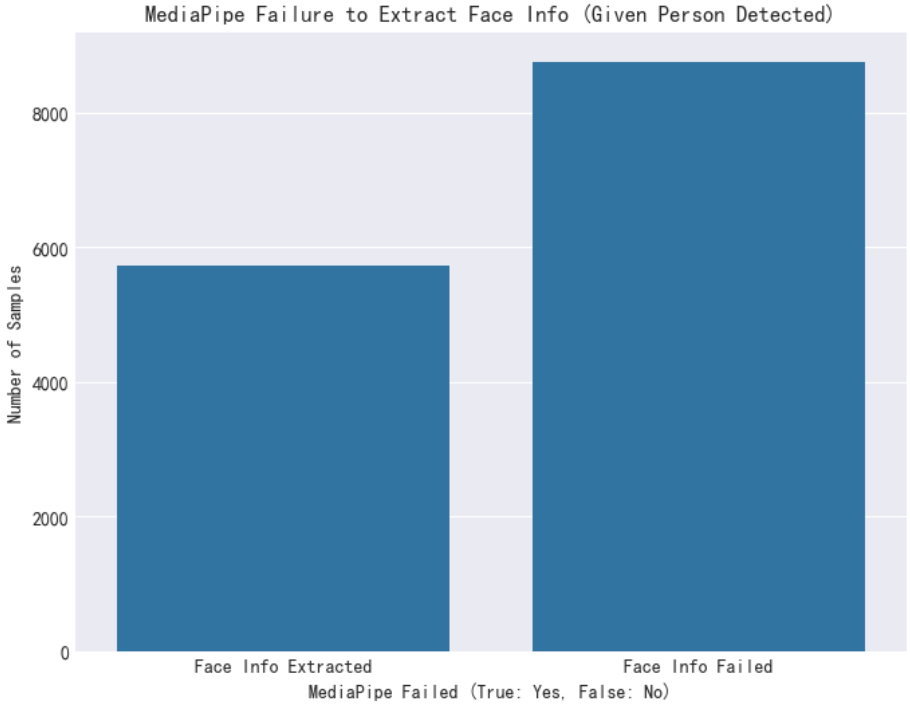
Use Mediapipe to extract feature from ‘person’





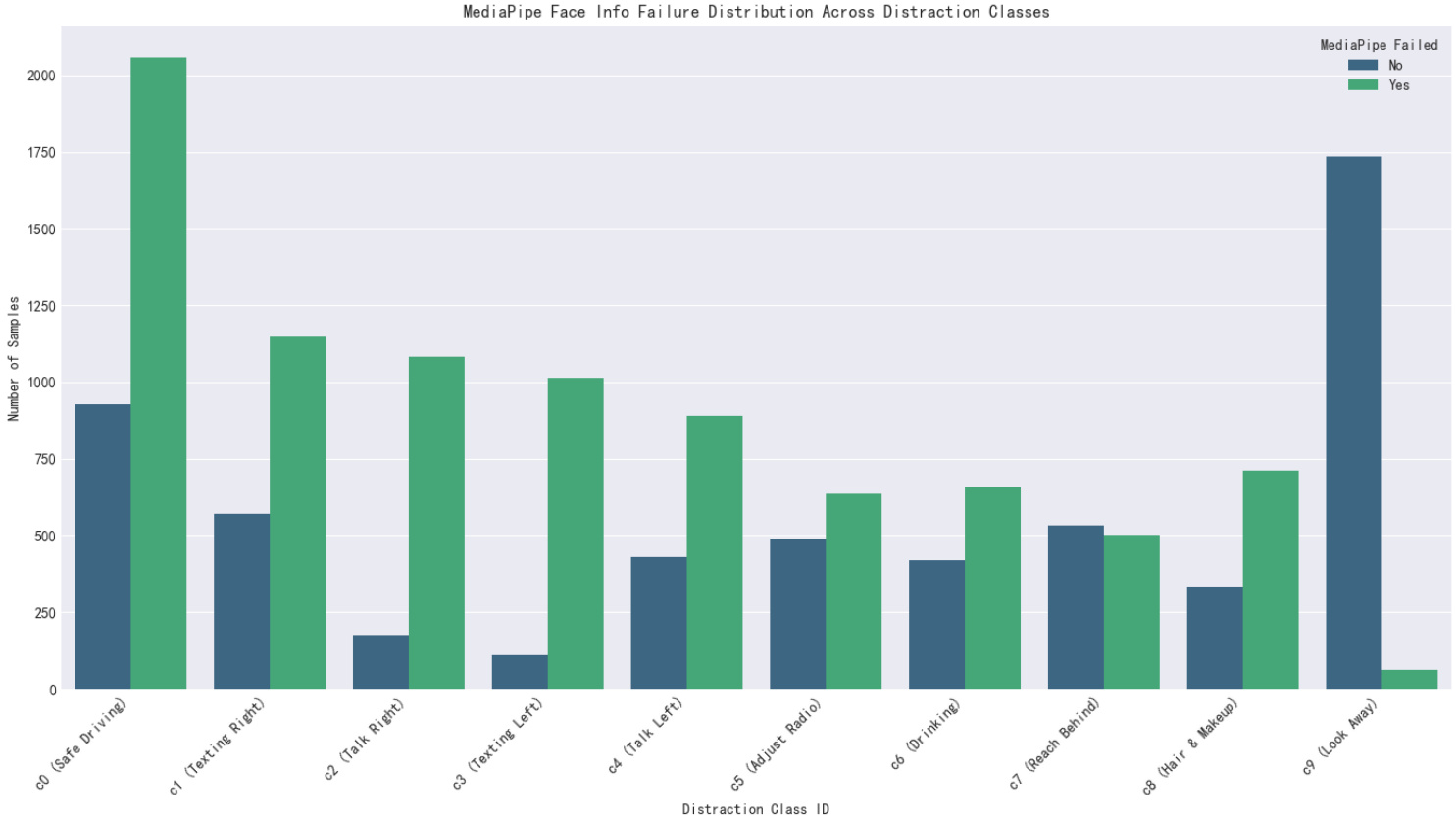


Use Mediapipe to extract feature from ‘person’



40% successfully extract the facial feature

MediaPipe summary



C0 Pic example

c0 (Safe Driving): This class shows a significantly taller green bar, indicating that for most "Safe Driving" samples, MediaPipe failed to extract meaningful face information. This might be due to drivers maintaining a very straight head pose, which offers less distinctive movement for pose estimation, or perhaps subtle variations in image quality or angle for "standard" faces.

C3 Pic example

c3 (Texting Left): The green bar is much taller than the blue bar, suggesting a high failure rate for MediaPipe in extracting pose/EAR in the "Texting Left" category. This could be attributed to the driver looking down at their phone, leading to an unfavorable face angle or partial occlusion, making face landmark detection challenging.

C7 Pic example

c7 (Reach Behind): This class shows a mixed distribution, with the green bar being slightly taller, implying more failures than successes. "Reaching Behind" actions often involve significant head rotation or body contortion, which might push the face into extreme angles that MediaPipe struggles to process accurately.

C9 Pic example

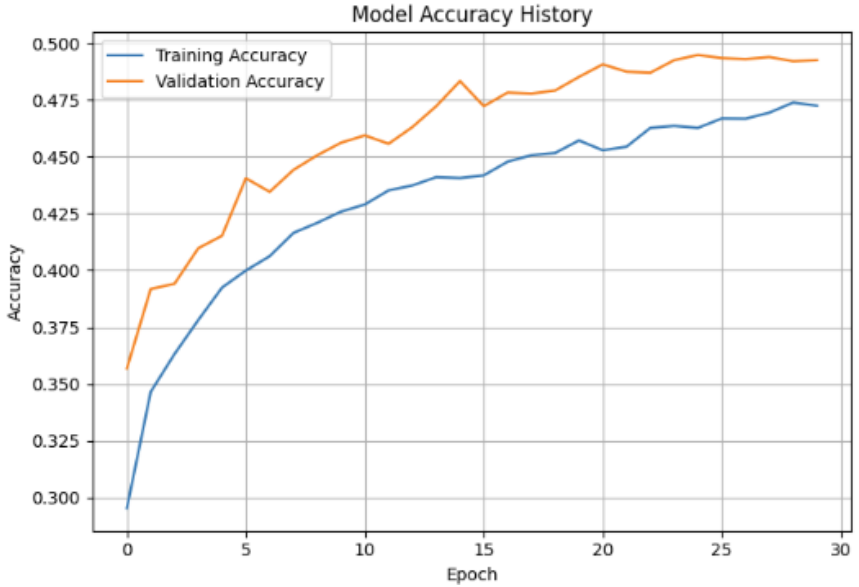
c9 (Look Away): This class has a very tall blue bar and a very small green bar. This is a positive sign, as it indicates that for the "Look Away" behavior, MediaPipe largely succeeded in extracting face information (Head Pose/EAR). This is reasonable, as looking away often involves a distinct head turn, providing clear signals for pose estimation.

MLP

```
model_classifier = keras.Sequential([
    layers.Input(shape=(input_dim,)),
    layers.Dense(128, activation='relu'),
    layers.Dropout(0.3),
    layers.Dense(64, activation='relu'),
    layers.Dropout(0.3),
    layers.Dense(num_classes, activation='softmax')
```

```
model_classifier.compile(optimizer='adam',
                         loss='sparse_categorical_crossentropy',
                         metrics=['accuracy'])
```

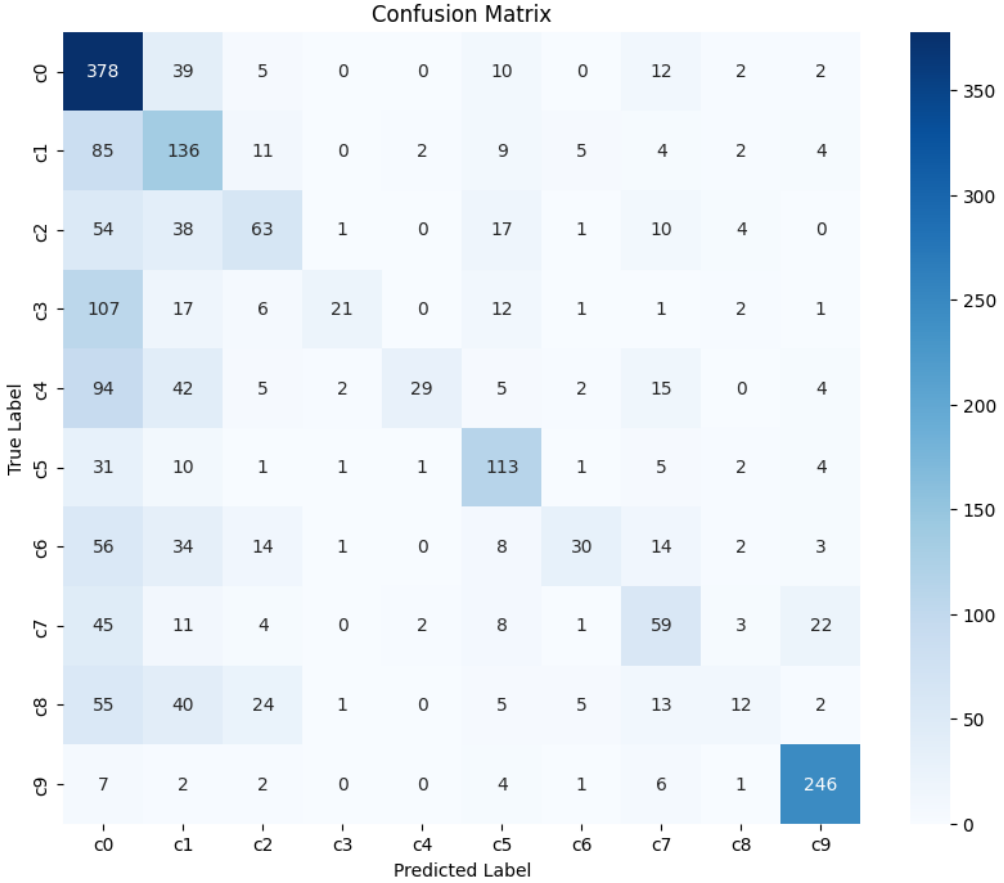
MLP Output



Classification Report:

	precision	recall	f1-score	support
c0	0.41	0.84	0.56	448
c1	0.37	0.53	0.43	258
c2	0.47	0.34	0.39	188
c3	0.78	0.12	0.22	168
c4	0.85	0.15	0.25	198
c5	0.59	0.67	0.63	169
c6	0.64	0.19	0.29	162
c7	0.42	0.38	0.40	155
c8	0.40	0.08	0.13	157
c9	0.85	0.91	0.88	269
accuracy			0.50	2172
macro avg	0.58	0.42	0.42	2172
weighted avg	0.57	0.50	0.46	2172

Confusion Matrix



Eye Aspect Ratio (EAR) Analysis

```
--- EAR (Eye Aspect Ratio) Analysis ---

EAR Mean and Standard Deviation per Class (excluding 0.0 defaults):
                ear_left           ear_right
                    mean       std      mean       std
original_class
0               0.805328  0.458326  0.234106  0.117585
1               0.772831  0.414353  0.181829  0.078133
2               0.259017  0.157524  0.490476  0.217356
3               0.489124  0.401833  0.391658  0.230019
4               0.734475  0.414262  0.233476  0.098479
5               0.628960  0.337333  0.225916  0.171999
6               0.724746  0.530793  0.243308  0.137375
7               0.350127  0.351589  0.260960  0.157233
8               0.584369  0.332931  0.316093  0.164273
9               0.255528  0.116320  0.253484  0.086368
```

Conclusion

In this project, we successfully developed and implemented a comprehensive deep learning pipeline for multi-class driver distraction detection. Our key achievements include:

- **Robust Feature Extraction:** We built a multi-stage feature extraction system that effectively combines:
  - **YOLOv8:** For accurate object detection (person, cellphone, bottle/cup) and generating pseudo-labels.
  - **MediaPipe Face Mesh:** For precise facial landmark detection, enabling the calculation of Eye Aspect Ratio (EAR) and 3D Head Pose (Pitch, Yaw, Roll angles).
- **Feature Analysis & Insights:** We performed detailed analysis of the extracted features, including correlation heatmaps and distribution plots. This revealed meaningful relationships between features and highlighted the subtle behavior of EAR and head pose across different distraction categories.
- **Functional Classifier:** We successfully trained a MLP classifier using these combined features, capable of categorizing driver states into 10 distinct distraction classes (c0-c9).
- **Model Performance & Context:** The final classifier achieved a test accuracy of over **50%**. While this signifies a substantial improvement over random chance in a complex 10-class problem.

Thank you