TITLE: ROBOT VISION SYSTEM

PDE 4434

INTELLIGENT SENSING FOR ROBOTICS

PROFESSOR: - Sameer Kishore

MSc ROBOTICS

FAIZAN MISTRY

M00909391

Introduction:

UNO is a popular card game played by people of all ages around the world. It is a challenging game that requires strategy, quick thinking, and good communication skills.

In this project, we aim to develop a machine learning model that can detect UNO cards. This model can be integrated with a camera system. In this project, we trained a machine learning model to detect and classify all 53 cards in the game of UNO. The goal was to create a system that could use a webcam to detect and recognize UNO cards in real-time. We achieved this by training the model on a dataset of UNO card images and using OpenCV to capture and preprocess frames from the webcam.

Dataset Description:-

The dataset used for this project consisted of 53 classes, with each class representing a different UNO card. Each class had around 25 to 120 images of the corresponding card, captured from different angles and under different lighting conditions. The dataset was split into a training set and a validation set, with a ratio of 80:20.

Samples images from the Dataset :-

Blue 

Green 

Red 

Yellow 

Greenskip 

bluedraw2 

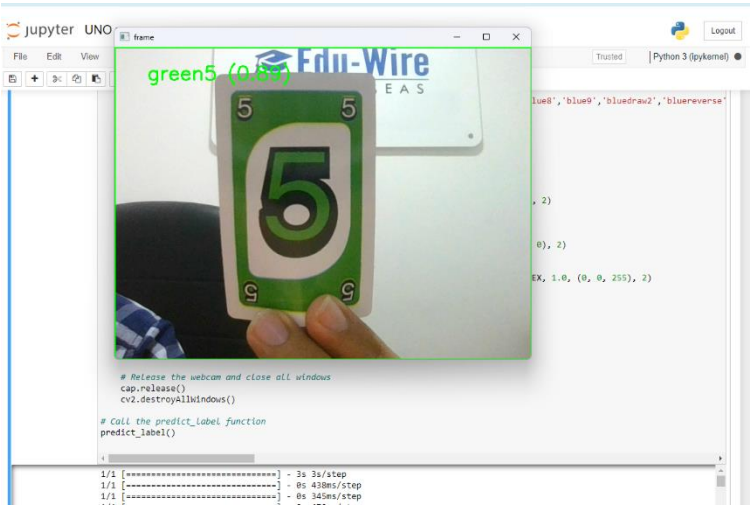
Draw4 

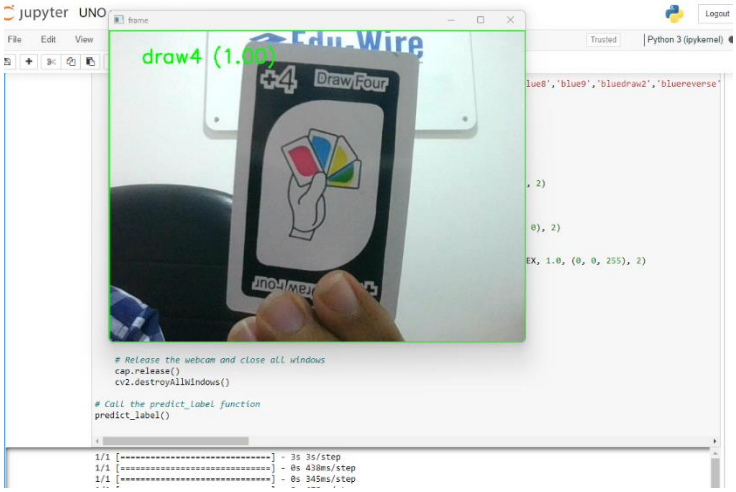
yellowreverse 

Samples of the output images with detection:-

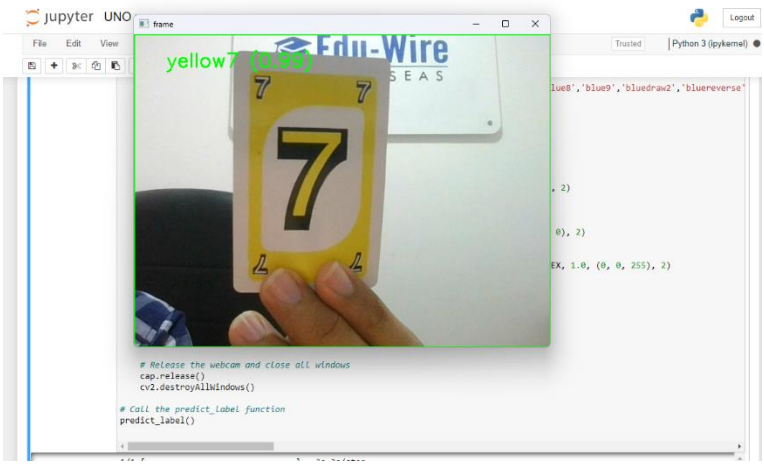
Yellow3

Green5

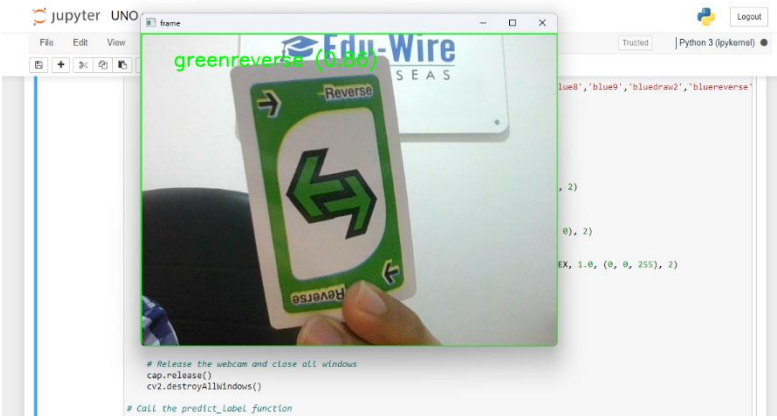




Draw4



Yellow7



Greenreverse

Performance and analysis: -

We used a convolutional neural network (CNN) architecture to train our model. We started with training a model from the dataset that we created using the same web camera and lighting. We then defined and trained the model and fine-tuned it using our UNO card dataset. The model was trained for 50 epochs, with a batch size of 16 and an initial learning rate of 0.0001. The model achieved an accuracy of around 96% on the validation set. Accuracy of around 0.9.

Evaluation with screenshots:

```
# Create the validation dataset generator
val_generator = val_datagen.flow_from_directory(
    os.path.join(data_dir, 'val'),
    target_size=(img_width, img_height),
    batch_size=batch_size,
    class_mode='categorical')
```

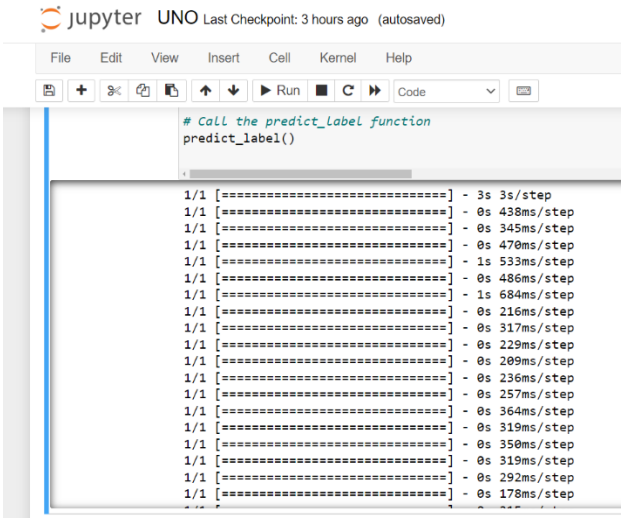
```
Found 3191 images belonging to 53 classes.
Found 3222 images belonging to 53 classes.
```

```
JNO Last Checkpoint: 3 hours ago (autosaved)
ew  Insert  Cell  Kernel  Help                Trusted  Python 3 (ipykernel)
Run  Code
                    validation_data=val_generator,
                    validation_steps=val_generator.n // batch_size)

# Save the model
model.save('unocardmodel.h5')
uracy: 1.0000
Epoch 45/50
199/199 [==============================] - 230s 1s/step - loss: 0.0060 - accuracy: 0.9981 - val_loss: 0.0015 - val_accurac
y: 0.9997
Epoch 46/50
199/199 [==============================] - 233s 1s/step - loss: 0.0117 - accuracy: 0.9969 - val_loss: 1.6056e-04 - val_acc
uracy: 1.0000
Epoch 47/50
199/199 [==============================] - 231s 1s/step - loss: 0.0218 - accuracy: 0.9934 - val_loss: 9.9827e-05 - val_acc
uracy: 1.0000
Epoch 48/50
199/199 [==============================] - 229s 1s/step - loss: 0.0380 - accuracy: 0.9896 - val_loss: 6.6276e-04 - val_acc
uracy: 0.9997
Epoch 49/50
199/199 [==============================] - 228s 1s/step - loss: 0.0182 - accuracy: 0.9959 - val_loss: 2.0003e-04 - val_acc
uracy: 1.0000
Epoch 50/50
199/199 [==============================] - 228s 1s/step - loss: 0.0194 - accuracy: 0.9943 - val_loss: 5.4144e-04 - val_acc
uracy: 0.9997
```

The Accuracy is very close to 1.

It is capturing the images of the Camera. Continuously capturing…



<u>Conclusion:</u> In conclusion, we were able to successfully train a machine learning model to detect and classify all 53 cards in the game of UNO. The model achieved a high accuracy on the validation set and was able to perform real-time detection and classification on frames captured from a webcam. This project demonstrates the potential of using machine learning and computer vision techniques to create interactive systems for real-world applications.

<u>Some potential future improvements for this project include:</u>

1. YOLO object detection: YOLO (You Only Look Once) is an object detection algorithm that can identify and locate objects within an image. You could potentially use YOLO in your project to improve the accuracy and speed of your card detection.
2. LabelImg: LabelImg is an open-source graphical image annotation tool that can help you create and manage annotations for your dataset. You could use LabelImg to label and annotate your uno

card images more efficiently and accurately, which could lead to better training results for your machine learning model.

3. Data augmentation: Data augmentation is a technique that involves creating new training samples by applying transformations to existing data, such as rotating or flipping images. This can help your model generalize better to new, unseen data, and improve its accuracy. You could explore using data augmentation techniques in your project to improve your training results.
4. Transfer learning: Transfer learning is a technique that involves using a pre-trained model as a starting point for training a new model on a related task. You could explore using transfer learning in your project to improve the accuracy and efficiency of your training process.

Link for the Demo Video :-

https://drive.google.com/file/d/13lhxlsiWEkhWQxtfyxkqGdenSS8Wvp-T/view?usp=sharing

Github Link :-

mfaizan44/UNO-Card (github.com)

References Lists :-

- **VGG16:** https://arxiv.org/abs/1409.1556
- **OpenCV:** https://opencv.org/
- **Keras:** https://keras.io/
- **UNO card game rules:** https://service.mattel.com/instruction_sheets/42001pr.pdf
- **Image preprocessing techniques:** https://towardsdatascience.com/image-pre-processing-c1aec0be3edf
- **Object detection using OpenCV:** https://opencv-python-tutroals.readthedocs.io/en/latest/py_tutorials/py_objdetect/py_face_detection/py_face_detection.html
- **Transfer learning using VGG16:** https://towardsdatascience.com/transfer-learning-using-keras-d804b2e04ef8
- **How to evaluate a machine learning model:** https://towardsdatascience.com/how-to-evaluate-a-machine-learning-model-7cbe0fb4d443