To,

IITD-AIA Foundation of Smart Manufacturing

Subject: **Weekly Progress Report for Week 2**

Dear Sir,

Based on my understanding and the topics covered, I have prepared the following progress report that addresses the relevant objectives of the project.

What happened last week – Week1 :

- Deep Learning in Computer Vision Basics
- Convolution operation, padding, strided convolutions, and convolutions over three-dimensional volumes.
- Learned about OpenCV and Pillow.
- Image manipulation with Pillow and OpenCV, Pixel Transformations, Geometric Operations.
- Tenosrflow Data Input Pipeline.

What's happening this -week2:

- Neural Network Architecture
- Model building
- Convolutional Neural Network Classifier
- Forward propagation, Gradient descent, Loss function, Activation Function
- Learned about Keras and Tensorflow libraries
- Implementation of OpenCV

Weekly Progress:

June 12:(Monday)

I dedicated my time to studying Neural Network architectures and the process of building models.

To solidify my understanding, I applied a Convolutional Neural Network (CNN) model to a sample dataset .

June 13:(Tuesday)

I deepened my understanding of Convolutional Neural Network (CNN) classifiers.

Additionally, I ventured into the realm of deploying models using CV Studio. Through this exploration, I gained valuable insights into the process of deploying a CNN model for practical applications.

June 14:(Wednesday)

Learned about KNN for object recognition. KNN can be used for image classification tasks.

Given a new image, KNN classifies it by comparing it with the labelled images in the training dataset.

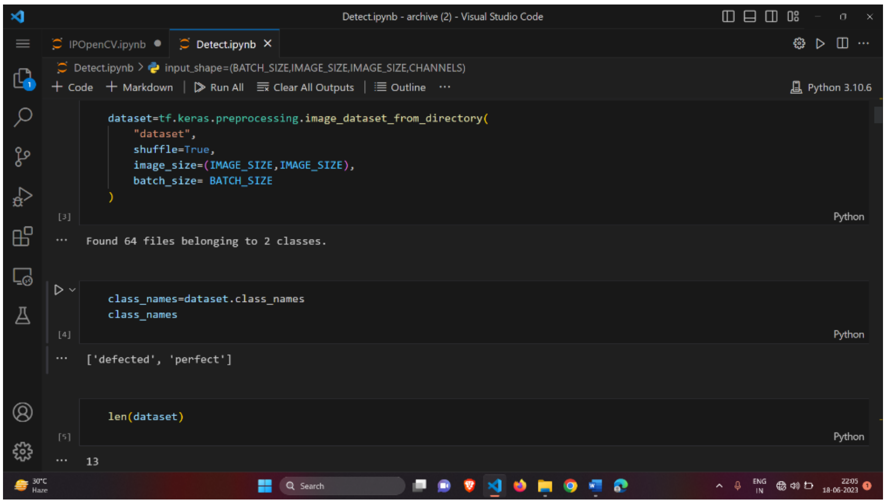
It calculates the distance between the new image and the training images and assigns the class label based on the majority vote of the K nearest neighbours.

June 15:(Thursday)

I acquired knowledge about fundamental concepts in deep learning, such as forward propagation, gradient descent, and loss functions. Additionally, I familiarized myself with the main steps involved in every Deep Learning model.

To apply my understanding, I created image batches and visualized the plotted images.

Moreover, I practiced calculating activations, deepening my comprehension of the concepts.



Detect.ipynb - archive (2) - Visual Studio Code
IPOpenCV.ipynb
Detect.ipynb
Detect.ipynb > input_shape=(BATCH_SIZE,IMAGE_SIZE,IMAGE_SIZE,CHANNELS)
+ Code + Markdown | Run All Clear All Outputs | Outline ...
Python 3.10.6

```
dataset=tf.keras.preprocessing.image_dataset_from_directory(
    "dataset",
    shuffle=True,
    image_size=(IMAGE_SIZE,IMAGE_SIZE),
    batch_size= BATCH_SIZE
)
```

[3]

```
Found 64 files belonging to 2 classes.
```

```
class_names=dataset.class_names
class_names
```

[4]

```
['defected', 'perfect']
```

```
len(dataset)
```

[5]

```
13
```

30°C Haze
Search
ENG IN
22:05 18-06-2023

June 16:(Friday)

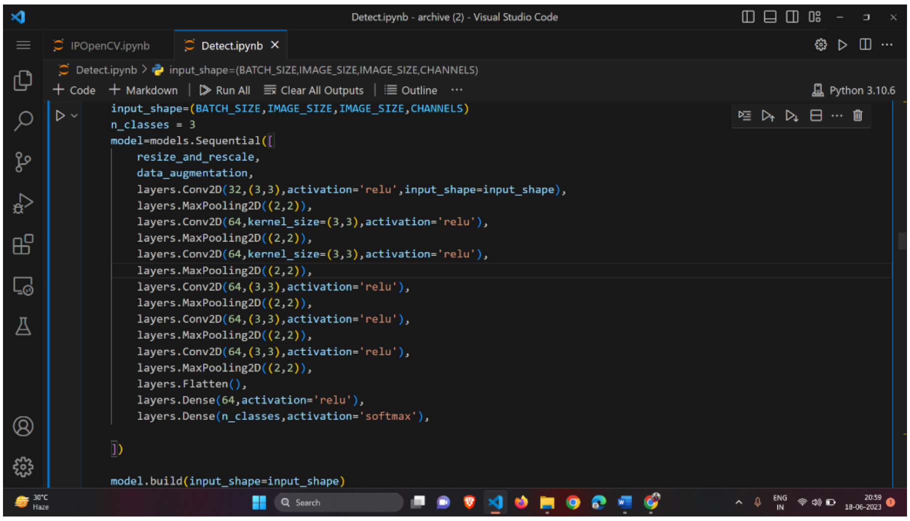
I revised the fundamental concepts of Deep Learning, delved into OpenCV, and gained insights into performing explanatory data analysis.

Today, I put my learnings from yesterday into practice by implementing the concepts.

June 17:(Saturday)

Today, my focus was on understanding and practicing model building using Convolutional Neural Networks (CNN).

I extensively explored the capabilities of the Keras and TensorFlow libraries. Putting my knowledge into action, I successfully built a CNN model tailored to a sample dataset.



```
input_shape=(BATCH_SIZE,IMAGE_SIZE,IMAGE_SIZE,CHANNELS)
n_classes = 3
model=models.Sequential([
    resize_and_rescale,
    data_augmentation,
    layers.Conv2D(32,(3,3),activation='relu',input_shape=input_shape),
    layers.MaxPooling2D((2,2)),
    layers.Conv2D(64,kernel_size=(3,3),activation='relu'),
    layers.MaxPooling2D((2,2)),
    layers.Conv2D(64,kernel_size=(3,3),activation='relu'),
    layers.MaxPooling2D((2,2)),
    layers.Conv2D(64,(3,3),activation='relu'),
    layers.MaxPooling2D((2,2)),
    layers.Conv2D(64,(3,3),activation='relu'),
    layers.MaxPooling2D((2,2)),
    layers.Conv2D(64,(3,3),activation='relu'),
    layers.MaxPooling2D((2,2)),
    layers.Flatten(),
    layers.Dense(64,activation='relu'),
    layers.Dense(n_classes,activation='softmax'),

])

model.build(input_shape=input_shape)
```

June 18:(Sunday)

I implemented various functionalities of OpenCV, including converting RGB images to grayscale, resizing images, flipping images, cropping images, and drawing shapes such as rectangles, circles, and lines on the images.

Resized Image

Cropped Image

Flipped Image

Drawn Circle

On the image