MACHINE LEARNING

ASSIGNMENT - 04

**NAME ROLL NUMBER SECTION**

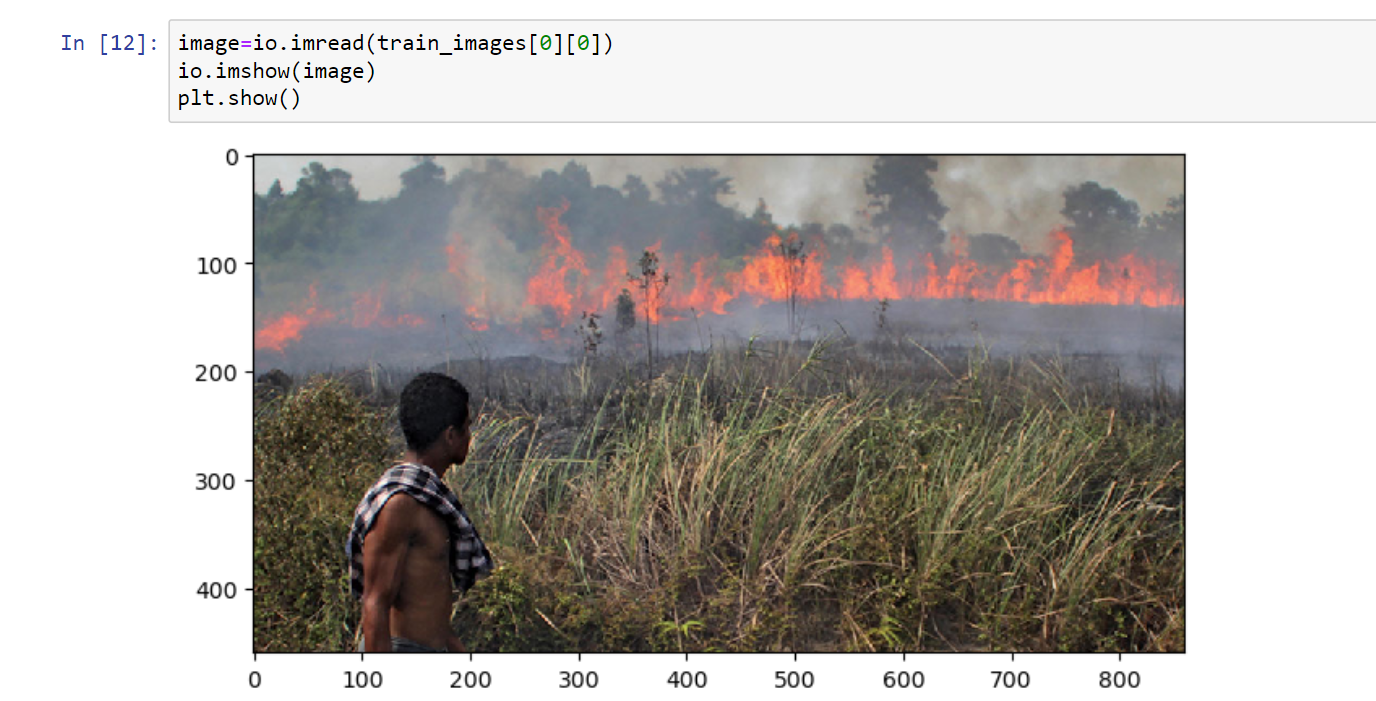
Zaraar Malik 21i-2705 BS-AI-5K

**MACHINE LEARNING PROJECT REPORT**

**Dataset Description :**

The dataset that has been used in my ML project includes images that represent Wild Fire and images that do not have Fire in it. The dataset is called Wild Fire Detection Dataset. This is a public dataset and the best thing about the dataset is that it can be extended to however we see fit . The reason for that is there are multiple Datasets that have fire and no-fire images each which target a specific problem. So , we can extend our dataset by adding more fire and no-fire images into it which in turn will make our ML model more accurate and robust however it will involve more trainning time . This problem is also taken care of in my ML project by Hyper-Parameter Tuning Grid-Search CV .

**SAMPLE IMAGE FROM THE DATASET**



**Strengths of My ML Project:**

So, basically my current ML project solves a very in-demand classification problem . So, the dataset has been used to pre-dict whether an image has wildfire or not by extracting SOTA Features and then Training a SOTA Classification Model such as a CNN itself attached with a Deep Neural Network.

My aim is to train a simple Neural Network but by extracting extremely resourceful features by Comprehensive Feature Engineering which will hep reduce the cost of Model Training while maintining the Robustnessa and Accuracy of the Model.

The following are the main Strengths of my ML Project :

* **Customized Feature Extraction:**In my approach, I manually extract statistical and fractional features, tailoring the feature set to the specific characteristics of wildfire images. This customization enhances my model's ability to capture domain-specific patterns and nuances related to fire detection, leveraging my knowledge of the domain.This targeted feature extraction allows for a more nuanced understanding of wildfire imagery, ensuring that the model focuses on key aspects crucial for accurate detection.
* **Incorporation of Domain-Specific Visual Cues:** I integrate domain expertise into the feature extraction process, enabling the inclusion of features that are specifically relevant to the unique visual cues associated with wildfires. This incorporation ensures that the selected features align closely with the distinctive aspects of wildfire imagery, potentially improving my model's discriminative power. The integration of domain-specific visual cues not only enhances the model's accuracy but also provides valuable insights into the decision-making process. This approach enables a more informed selection of features, ensuring that the model focuses on capturing the subtle visual characteristics indicative of wildfire instances.
* **Simplicity for Deployment and Interpretability:**I utilize a simple neural network with 5 layers, providing a straightforward and interpretable model. The simplicity of the architecture facilitates ease of understanding, interpretation, and deployment, making it valuable for stakeholders who require transparency in the decision-making process of my wildfire detection system. The simplicity of the model not only aids in deployment but also streamlines the model's maintenance and updates. Stakeholders can easily comprehend the model's decisions, fostering trust and facilitating collaborative decision-making in wildfire management.
* **Robustness to Noisy Environments:** My approach demonstrates robustness to noisy environmental conditions commonly encountered in wildfire scenarios. The manual extraction of features, combined with a simplified neural network, allows the model to focus on essential visual cues, making it more resilient to variations in lighting, weather, or other potential sources of noise. The model's robustness enhances its reliability in diverse environmental conditions, ensuring consistent performance. This robustness is particularly crucial in real-world applications, where the model needs to operate effectively under varying circumstances for timely and accurate wildfire detection.

**Weaknessof ML Project:**

Even though the a lot of effort has been instilled in this project , however there are still some drawbacks of my ML project , since I have removed the SOTA CNN model so there must be some catch where my approach would not be able to compete with the SOTA models .

The following are some drawbacks of my ML projetct :

* **Limited Capacity for Hierarchical Feature Learning:**Manual feature extraction may limit the model's capacity to learn hierarchical representations of data. Deep learning models, with their ability to automatically learn complex hierarchical features, could outperform manually extracted features in capturing intricate patterns relevant to wildfire detection.The model's ability to discern subtle hierarchical features, which may be critical for distinguishing between different wildfire instances, might be constrained.
* **Dependency on Feature Quality:** The success of my model heavily relies on the quality and relevance of the manually extracted statistical and fractional features. If these features do not adequately represent the distinctive characteristics of wildfire and non-wildfire instances, the model's classification performance may be compromised.Ensuring high-quality feature extraction is essential for the model's effectiveness, and careful consideration of feature selection is crucial.
* **Potential Overfitting Concerns:**The simplicity of the neural network architecture, combined with manual feature extraction, may increase the risk of overfitting, especially if the model is not adequately regularized. Overfitting occurs when the model performs well on the training data but fails to generalize to unseen data.Implementing effective regularization techniques and thorough model evaluation are critical to mitigate the risk of overfitting.
* **Sensitivity to Input Variability:** The model's performance may be sensitive to variations in environmental conditions, lighting, or other factors, especially if the manually extracted features do not adequately account for these variations. This sensitivity could lead to reduced robustness in diverse scenarios.Incorporating features that capture variability and employing techniques to enhance the model's robustness to diverse conditions may help mitigate this weakness.
* **Challenges in Handling Complex Spatial Patterns:**Manual feature extraction may struggle to capture complex spatial patterns in wildfire images, which are crucial for accurate detection. Unlike deep learning models designed for image-related tasks, your approach might not effectively recognize intricate spatial features. Ensuring that the selected features adequately represent the spatial relationships in the images and exploring additional techniques to capture spatial dependencies may be necessary to address this challenge.

**MY PERSONAL CONTRIBUTION TO THE PROJECT**

In this project, my significant contribution lies in the thoughtful selection and extraction of three vital features for effective wildfire detection. The Following are the 3 main features that I have extracted and made a single feature vector via feautre crossing :

* **Histogram of Oriented Gradients (HOG):** Captures edges and gradients in images, crucial for recognizing patterns like smoke and flames in wildfire pictures.
* **Histogram Features:** Detects changes in color intensity, providing the model with insights into variations that can indicate the presence of wildfires under different environmental conditions.
* **Local Binary Pattern (LBP) :** Focuses on texture differences in images, aiding the model in identifying unique visual cues associated with wildfires, such as the texture of smoke and flame patterns.

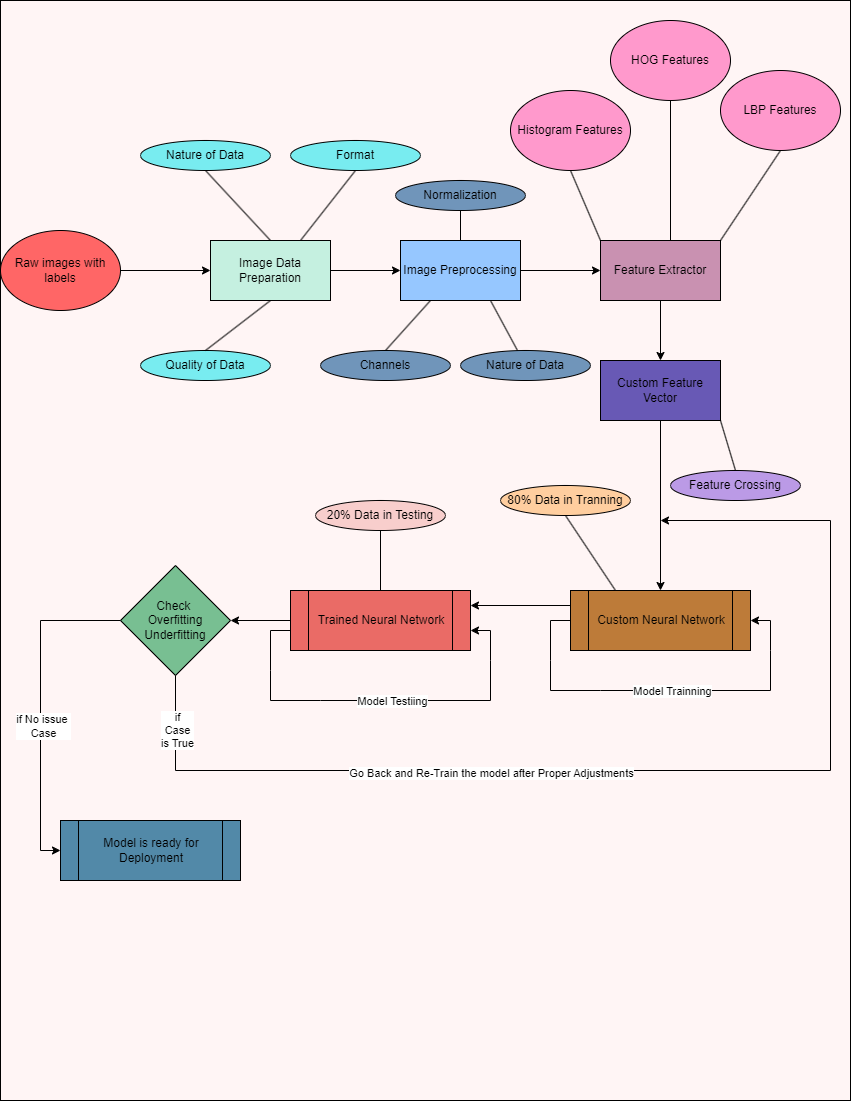
By incorporating HOG features, the model gains the ability to recognize edges and gradients, essential for capturing patterns indicative of flames. The inclusion of Histogram features enables the model to detect changes in color intensity, adding a valuable dimension to the feature set. Additionally, the incorporation of Local Binary Pattern features allows the model to focus on texture variations in wildfire imagery, enhancing its ability to detect unique visual cues associated with wildfires. With this contribution I haved opened the doors towards complex feature engineering to achieve high results in training ML model .

**Hyper-Parameters To Consider :**

As I would be trainnig a Neural Network , the Hyper-Parameters that I would be considereing are the following :

1. Learning Rate
2. Number of Neurons in each Layer
3. Number of layers in the NN ( I will try to keep max 5 layers)
4. Type of Loss Function
5. Type of Optimizer

**HIGH LEVEL ARCHITECTURE OF THE PROJECT**

****

**TASK-02**

**KEY STANDARD LIBRARIES:**

1. Numpy
2. Pandas
3. Skimage
4. Torch
5. Matplotlib
6. Sklearn

**FUNCTIONS OF KEY STANDARD LIBRARIES:**

1. **NumPy:**

* np.array: Creates a NumPy array.
* np.resize: Resizes the array.
* np.histogram: Computes the histogram of a set of data.
* np.astype: Converts the data type of the array.
* np.sum: Computes the sum of array elements.
* np.float32: Converts array elements to 32-bit floating-point.
* np.float64: Converts array elements to 64-bit floating-point.
* np.ravel: Flattens the array.

1. **Pandas:**

* pd.DataFrame: Creates a Pandas DataFrame.
* pd.head: Returns the first n rows of the DataFrame.
* pd.tail: Returns the last n rows of the DataFrame.

1. **Scikit-Image (skimage):**

* io.imread: Reads an image from a file.
* imshow: Displays an image.
* hog feature: Computes Histogram of Oriented Gradients for feature extraction.
* Local Binary Pattern: Computes Local Binary Pattern features.

1. **PyTorch:**

* Utils.data import dataset: Base class for PyTorch datasets.
* Utils.data import dataloader: Combines a dataset and a sampler for efficient data loading.
* nn: Neural network module in PyTorch.
* nn.Module: Base class for all neural network modules.
* Optimizer: Optimizer algorithms for updating model parameters.
* Loss: Loss functions for training neural networks.
* Activation functions: Functions applied to introduce non-linearity (e.g., torch.relu).

1. **Matplotlib:**

* plt.plot: Plots a graph.
* plt.show: Displays the plot.
* plt.subplot: Creates a subplot.
* plt.xlabel: Sets the label for the x-axis.
* plt.ylabel: Sets the label for the y-axis.
* plt.title: Sets the title of the plot.

1. **Scikit-Learn (sklearn):**

* model\_selection import classification\_report: Generates a classification report.
* model\_selection import train\_test\_split: Splits the dataset into training and testing sets.
* metrics import confusion\_matrix: Computes a confusion matrix for classification models.

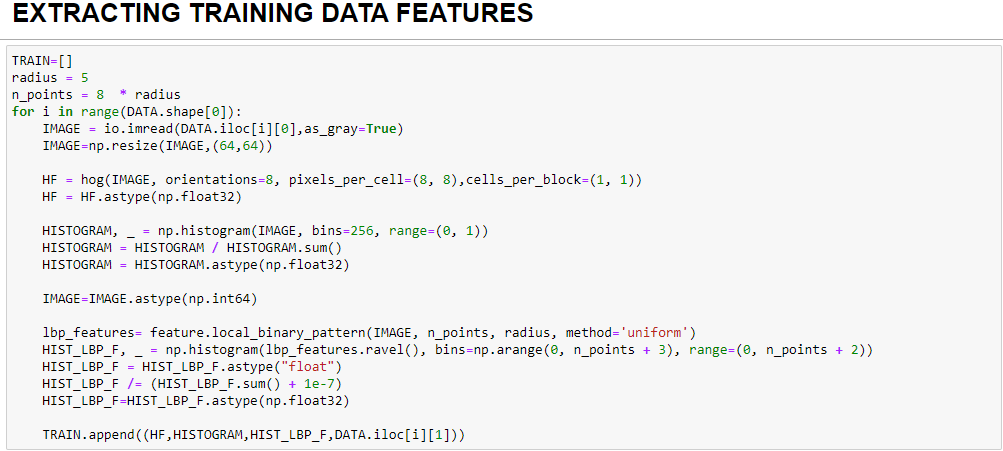
**HOW FUNCTIONS OF KEY STANDARD LIBRARIES HELP IN MY ML PROJECT**

Mainly there are 6 key standard libraries that have been used in my project . The First one is Numpy which is has been used to give the right shape and extract histogram features of the images. The second one is Pandas which is mainly used to store the paths of the image files and the labels of the image files . The Third one is Scikit-Image famously known as Skimage . I have used this library extensively for loading images and extracting Local Binary Pattern Features and Histogram of Oriented Gradients features from this library . Then comes the famous Pytorch Library that has been extensively used for creatiing a CustomDataset class as the dataset itself was very large. Secondly , it was used to create Custom Neural Network Class which was trained using the Trainning Data from the Dataset. It also includes selecting the specific Loss Functions , Activation Functions used in NN class and the Optimizer for the backward propagation . Futhermore , The Matplotlib library was used to visualize the different images and also after performing image-related operations , what are the resultant images . Finally , we have the sklearn library . This was mainly used for evaluations of the model such as displaying the Confusion Matrix , Classification Report and it was also used to split the data into Training and Testing part.

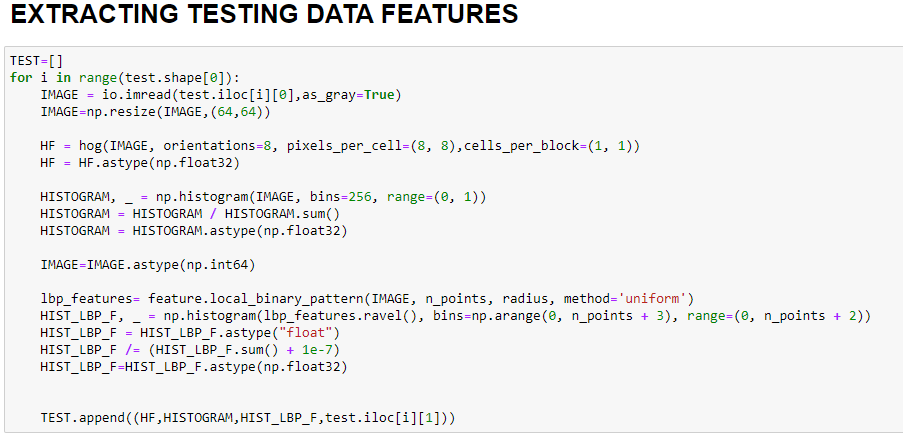
With this I conclude the assignment -04 of Machine Learning.

**CODE IMPLEMENTATION OF ML PROJECT**

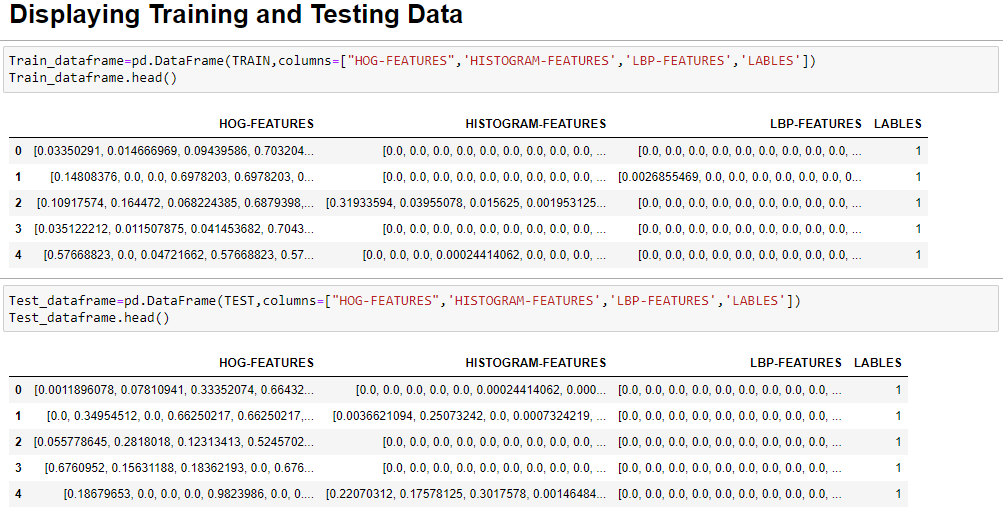
**READING AND LABELLING THE IMAGE DATA**

****

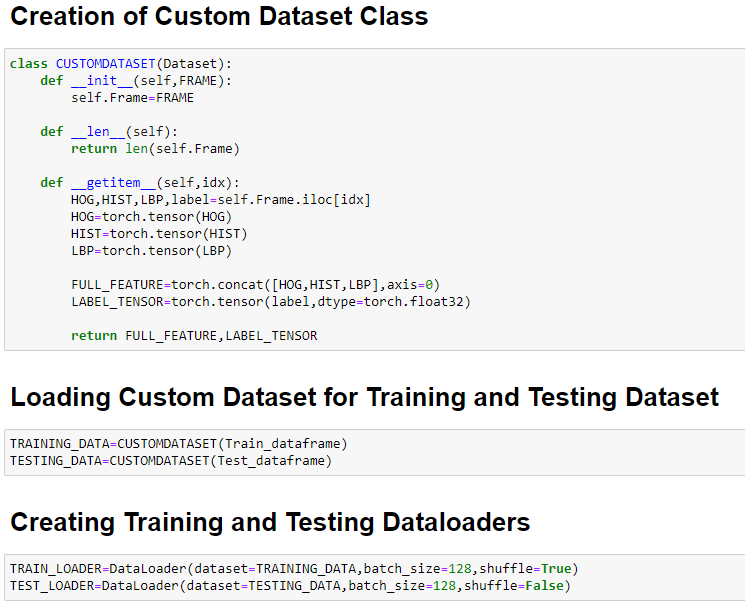
**EXTRACTING FEATURES OF THE DATASET**



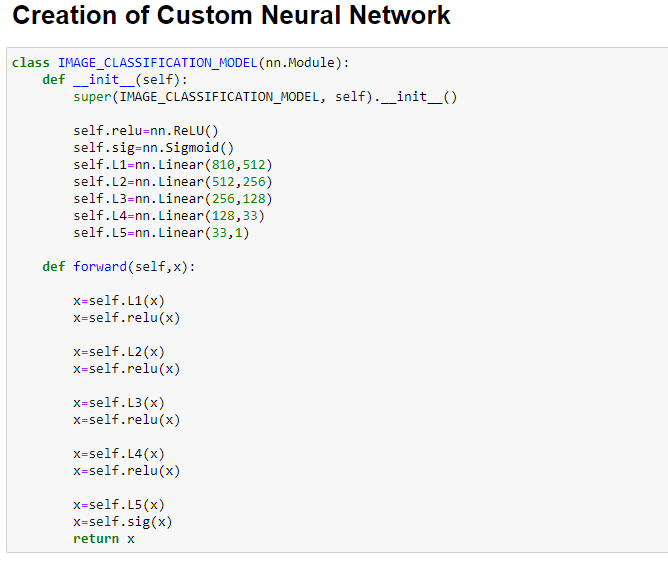
**DISPLAYING TRAINING AND TESTIING DATA**



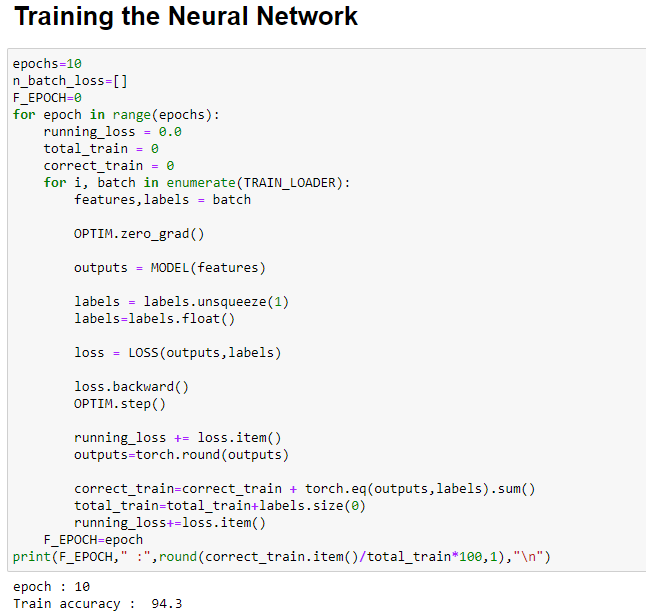
**CUSTOM DATASET CLASS , DATALOADERS**



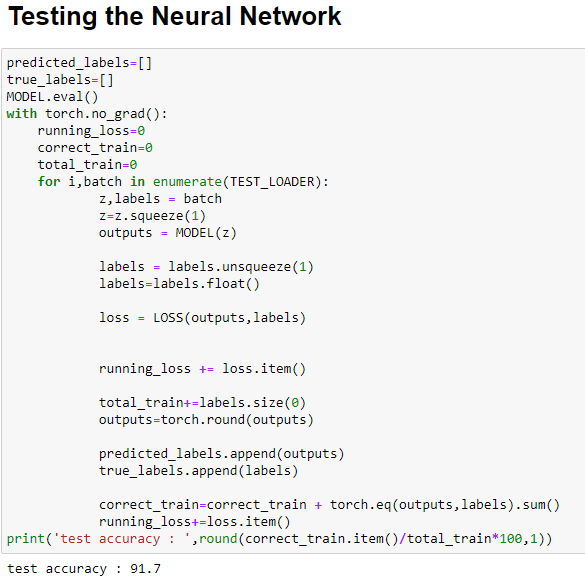
**CREATION OF CUSTOM NEURAL NETWORK CLASS**



**TRAINING THE NEURAL NETWORK**



**TESTING THE NEURAL NETWORK**



T

H Y

A O

N U

K