

Deep Learning

Mid-Course Test

edureka!

edureka!



Scenario 1

Problem Statement

The dataset consists of 2-Dimensional spectrograms of radio signals from space collected at the **SETI Institute** by the **Allen Telescope Array**. The objective is to classify the radio signals from outer space into one of four classes.



Dataset Description

SETI Dataset

- **Training Data:**
 - train_images: Normalized values of Pixels
 - train_labels: Stored as One-Hot Encoded data
- **Validation Data:**
 - val_images: Normalized values of Pixels
 - val_labels: Stored as One-Hot Encoded data
- **Classes:** “squiggle”, “narrowband”, “narrowbanddrd”, and “noise”



.ipynb file: Create a Notebook File and perform the following tasks

As a part of this test, you will be performing the following tasks:

- Prepare a detailed python notebook using CNN for classifying the radio signals from deep space using Keras from the SETI Dataset
- Import Required Libraries
- Load and Pre-process the dataset
 - Load the dataset using the pandas **read_csv** function
 - Check the shape of the training and validation data
 - Reshape the training and validation images
 - Hint: [Click Here!](#) to learn about NumPy function **reshape**
- Visualize the dataset
- Create Training and Validation Data Generators using Keras ImageDataGenerator function

Hint: [Click Here!](#) to learn about ImageDataGenerator

- Design a Convolutional Neural Network (CNN) Model
- Compile the Model using Adam optimizer, categorical_crossentropy loss function, and accuracy metric
- Print the Model summary
- Train the Model with batch_size = 32 & epochs = 12
- Evaluate the Model
 - Use the **model.evaluate** function to evaluate the accuracy
 - Print a **Classification Report** and the accuracy score (classification accuracy)
 - Display a **Confusion Matrix** to evaluate the performance of the model



Scenario 2

Problem Statement

The objective is to detect Moving Cars in a video file using OpenCV using the HaarCascade_car.xml file and then, you will use OpenCV to detect the License plates of a Car using the HaarCascade_russian_plate_number XML file



Tasks to be performed

.ipynb file 1: Create a Notebook File and perform the following tasks

As a part of this test, you will be performing the following tasks:

- Prepare a detailed python notebook using OpenCV to detect Moving Cars in a video
- Import Required Libraries
- Create a Classifier using the HaarCascade_car xml file
- Load the Video
- Detect the Moving Cars using the Classifier
- Display the Results



.ipynb file 2: Create a Notebook File and perform the following tasks

As a part of this test, you will be performing the following tasks:

- Prepare a detailed python notebook using OpenCV to detect the license plate of a car
- Import Required Libraries
- Load the Image using the **imread** function of OpenCV
- Pre-process the Image
 - Create a function to convert the image using `cv2.Color(img, cv2.COLOR_BGR2RGB)` function
 - Display the image
- Create a function for a classifier using the `HaarCascade_russian_plate_number.xml` file
- Detect the Car License Plates using the Classifier
- Display the Results

[Click Here!](#) To download the required files