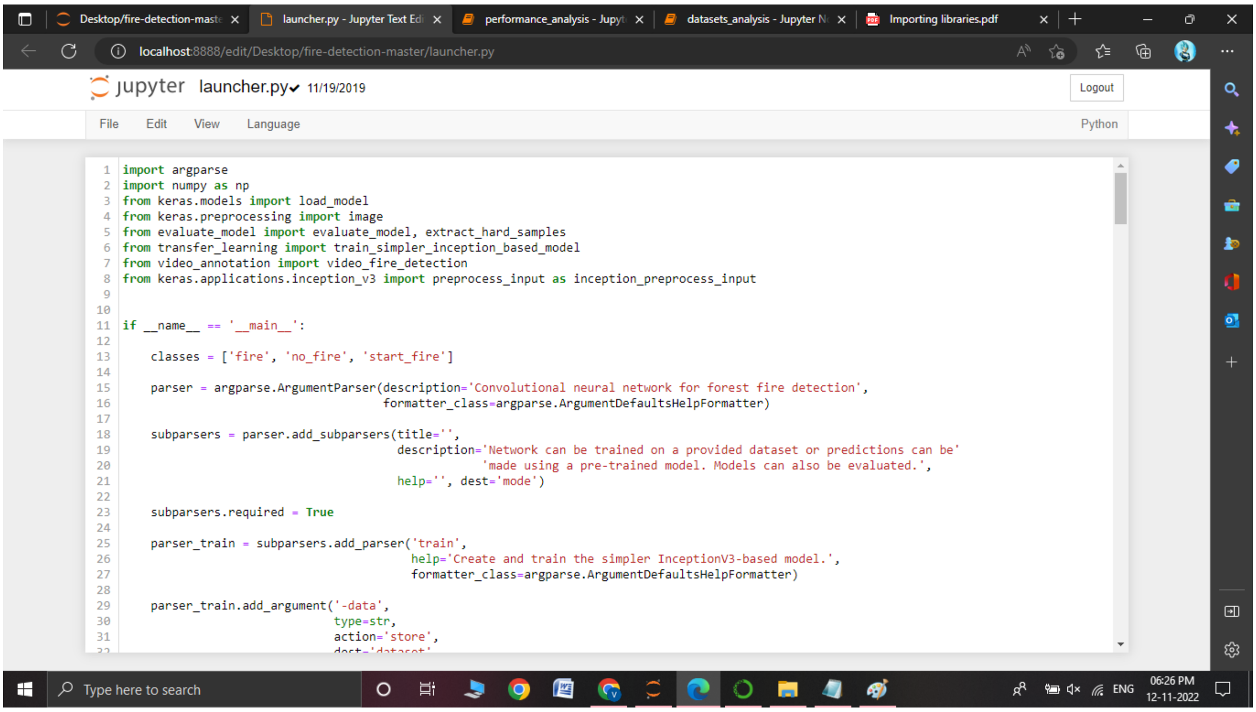
EMERGING METHODS FOR EARLY DETECTION OF FOREST FIRE

MODEL BUILDING

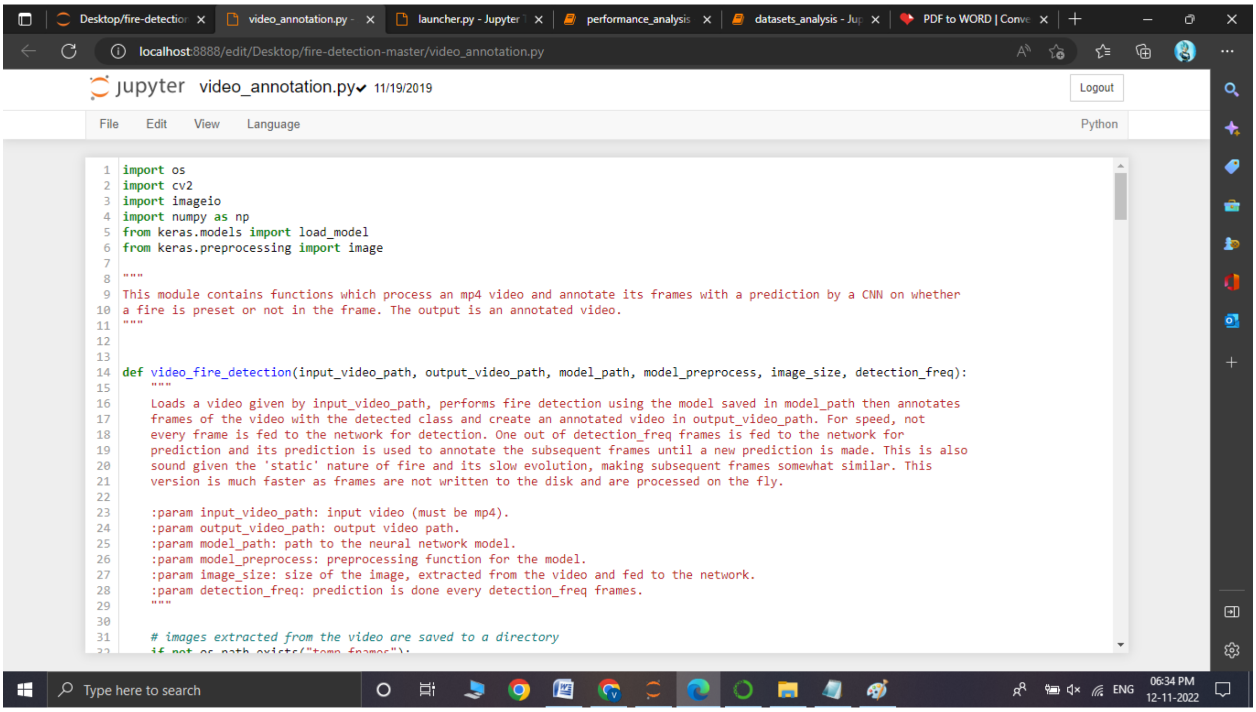
IMPORTING THE MODEL BUILDING LIBRARIES

Team ID	PNT2022TMID21968
Project Name	Project- Emerging methods for early detection of forest.

IMPORTING THE MODEL BUILDING LIBRARIES:


Desktop/fire-detection-maste
launcher.py - Jupyter Text Edi
performance_analysis - Jupyt
datasets_analysis - Jupyter N
Importing libraries.pdf
localhost:8888/edit/Desktop/fire-detection-master/launcher.py
jupyter
launcher.py
11/19/2019
Logout
File
Edit
View
Language
Python
Type here to search
ENG
06:26 PM
12-11-2022

```python
import argparse
import numpy as np
from keras.models import load_model
from keras.preprocessing import image
from evaluate_model import evaluate_model, extract_hard_samples
from transfer_learning import train_simpler_inception_based_model
from video_annotation import video_fire_detection
from keras.applications.inception_v3 import preprocess_input as inception_preprocess_input


if __name__ == '__main__':

    classes = ['fire', 'no_fire', 'start_fire']

    parser = argparse.ArgumentParser(description='Convolutional neural network for forest fire detection',
                                     formatter_class=argparse.ArgumentDefaultsHelpFormatter)

    subparsers = parser.add_subparsers(title='',
                                       description='Network can be trained on a provided dataset or predictions can be'
                                                   'made using a pre-trained model. Models can also be evaluated.',
                                       help='', dest='mode')

    subparsers.required = True

    parser_train = subparsers.add_parser('train',
                                         help='Create and train the simpler InceptionV3-based model.',
                                         formatter_class=argparse.ArgumentDefaultsHelpFormatter)

    parser_train.add_argument('-data',
                              type=str,
                              action='store',
                              dest='dataset'
```



```python
import os
import cv2
import imageio
import numpy as np
from keras.models import load_model
from keras.preprocessing import image

"""
This module contains functions which process an mp4 video and annotate its frames with a prediction by a CNN on whether
a fire is preset or not in the frame. The output is an annotated video.
"""


def video_fire_detection(input_video_path, output_video_path, model_path, model_preprocess, image_size, detection_freq):
    """
    Loads a video given by input_video_path, performs fire detection using the model saved in model_path then annotates
    frames of the video with the detected class and create an annotated video in output_video_path. For speed, not
    every frame is fed to the network for detection. One out of detection_freq frames is fed to the network for
    prediction and its prediction is used to annotate the subsequent frames until a new prediction is made. This is also
    sound given the 'static' nature of fire and its slow evolution, making subsequent frames somewhat similar. This
    version is much faster as frames are not written to the disk and are processed on the fly.

    :param input_video_path: input video (must be mp4).
    :param output_video_path: output video path.
    :param model_path: path to the neural network model.
    :param model_preprocess: preprocessing function for the model.
    :param image_size: size of the image, extracted from the video and fed to the network.
    :param detection_freq: prediction is done every detection_freq frames.
    """

    # images extracted from the video are saved to a directory
    if not os.path.exists("temp_frames"):
```