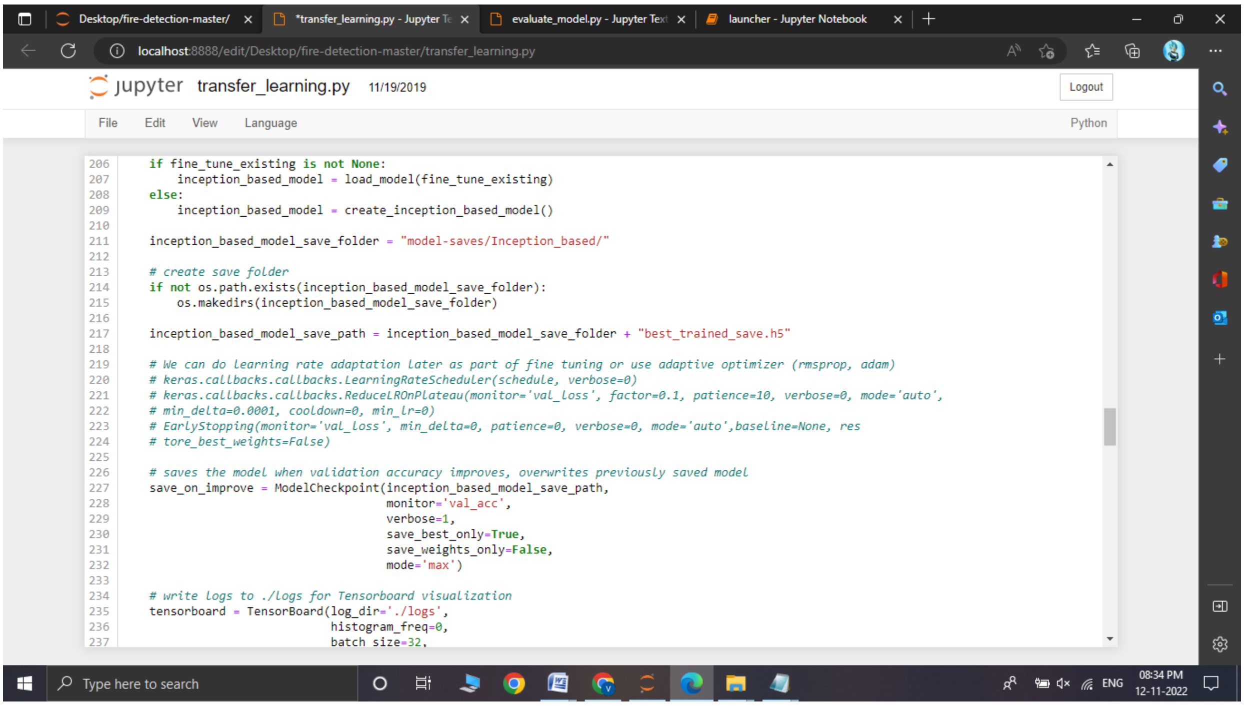
EMERGING METHODS FOR EARLY DETECTION OF FORESTFIRE

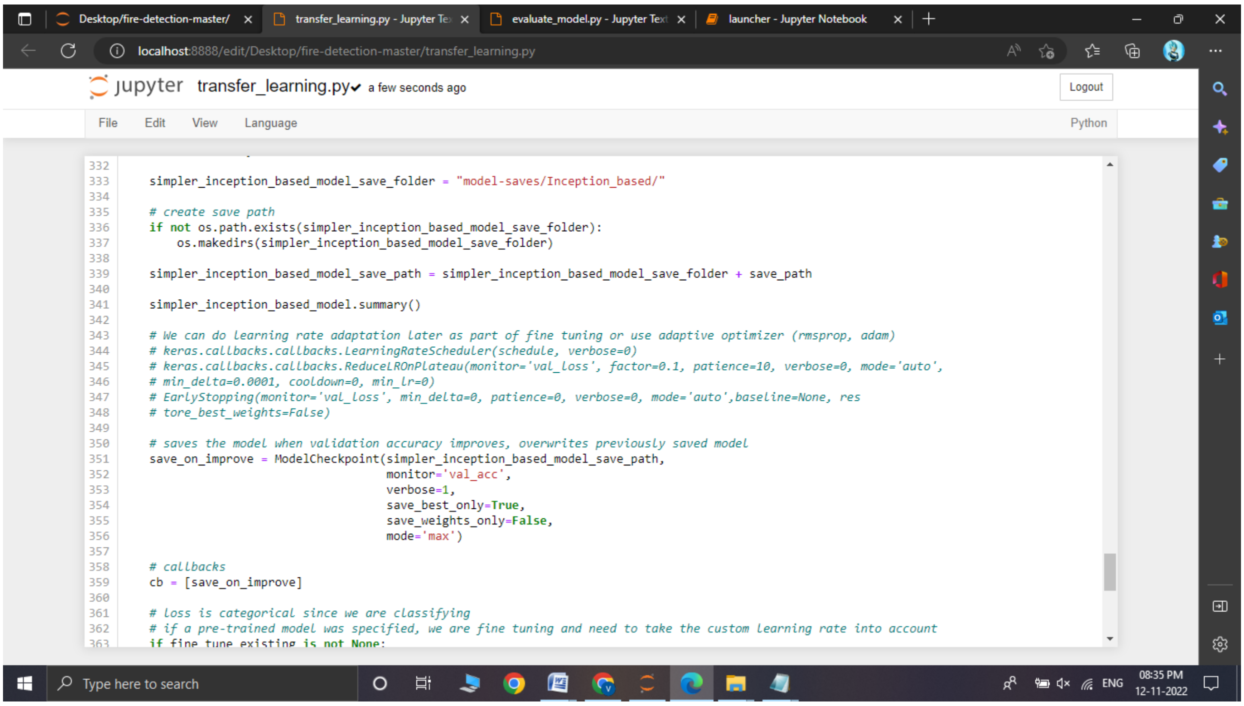
MODEL BUILDING

CONFIGURING LEARNING PROCESS

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


Desktop/fire-detection-master/
*transfer_learning.py - Jupyter Te
evaluate_model.py - Jupyter Text
launcher - Jupyter Notebook
localhost:8888/edit/Desktop/fire-detection-master/transfer_learning.py
jupyter transfer_learning.py 11/19/2019
Logout
File Edit View Language
Python
Type here to search
ENG
08:34 PM
12-11-2022

```
    if fine_tune_existing is not None:
        inception_based_model = load_model(fine_tune_existing)
    else:
        inception_based_model = create_inception_based_model()

    inception_based_model_save_folder = "model-saves/Inception_based/"

    # create save folder
    if not os.path.exists(inception_based_model_save_folder):
        os.makedirs(inception_based_model_save_folder)

    inception_based_model_save_path = inception_based_model_save_folder + "best_trained_save.h5"

    # We can do learning rate adaptation later as part of fine tuning or use adaptive optimizer (rmsprop, adam)
    # keras.callbacks.callbacks.LearningRateScheduler(schedule, verbose=0)
    # keras.callbacks.callbacks.ReduceLROnPlateau(monitor='val_loss', factor=0.1, patience=10, verbose=0, mode='auto',
    # min_delta=0.0001, cooldown=0, min_lr=0)
    # EarlyStopping(monitor='val_loss', min_delta=0, patience=0, verbose=0, mode='auto',baseline=None, res
    # tore_best_weights=False)

    # saves the model when validation accuracy improves, overwrites previously saved model
    save_on_improve = ModelCheckpoint(inception_based_model_save_path,
                                      monitor='val_acc',
                                      verbose=1,
                                      save_best_only=True,
                                      save_weights_only=False,
                                      mode='max')

    # write logs to ./logs for Tensorboard visualization
    tensorboard = TensorBoard(log_dir='./logs',
                              histogram_freq=0,
                              batch_size=32,
```


Desktop/fire-detection-master/
transfer_learning.py - Jupyter Te
evaluate_model.py - Jupyter Text
launcher - Jupyter Notebook
localhost:8888/edit/Desktop/fire-detection-master/transfer_learning.py
jupyter transfer_learning.py✔ a few seconds ago
Logout
File Edit View Language
Python
Type here to search
ENG
08:35 PM
12-11-2022

```
    simpler_inception_based_model_save_folder = "model-saves/Inception_based/"

    # create save path
    if not os.path.exists(simpler_inception_based_model_save_folder):
        os.makedirs(simpler_inception_based_model_save_folder)

    simpler_inception_based_model_save_path = simpler_inception_based_model_save_folder + save_path

    simpler_inception_based_model.summary()

    # We can do learning rate adaptation later as part of fine tuning or use adaptive optimizer (rmsprop, adam)
    # keras.callbacks.callbacks.LearningRateScheduler(schedule, verbose=0)
    # keras.callbacks.callbacks.ReduceLROnPlateau(monitor='val_loss', factor=0.1, patience=10, verbose=0, mode='auto',
    # min_delta=0.0001, cooldown=0, min_lr=0)
    # EarlyStopping(monitor='val_loss', min_delta=0, patience=0, verbose=0, mode='auto',baseline=None, res
    # tore_best_weights=False)

    # saves the model when validation accuracy improves, overwrites previously saved model
    save_on_improve = ModelCheckpoint(simpler_inception_based_model_save_path,
                                      monitor='val_acc',
                                      verbose=1,
                                      save_best_only=True,
                                      save_weights_only=False,
                                      mode='max')

    # callbacks
    cb = [save_on_improve]

    # loss is categorical since we are classifying
    # if a pre-trained model was specified, we are fine tuning and need to take the custom learning rate into account
    if fine_tune_existing is not None:
```