Department of Information Science and Technology

Anna University, Chennai – 25

IT5712 - PROJECT WORK (R2019) – Batch1 – Review 2

Academic Session: August 2022 – December 2022

One Page Writeup

Title of the Project: A DEEP-LEARNING APPROACH FOR DETECTING SPLICING & COPY-MOVE IMAGE FORGERIES AND IMAGE SELF-RECOVERY	
Team Members: Aravind J (2019115047), Krishnan S (2019115047), Pranay Varma (2019115067)	**Guide: Dr. K. Indra Gandhi**

Components used in the project that were existing:

CNN Approach: CNN convolutions definition with filter, max-pooling and activation layers (Referred a similar code as implementation of CNN in PyTorch is new for all of us).

Self-Consistency Learning: We have used a pre-trained model for checking consistency of the EXIF meta data as training the model required a high-end GPU.

Inbuilt functions available in Python and PyTorch have been used extensively in our code.

The module names written in the following section were coded by us

Team member wise, list the code sections written by him/her (source files, classes, functions etc.)

Krishnan S:

CNN Approach:

Module Patch Extractor:
Functions: check_and_reshape(), extract_all_patches(), delete_prev(), create_dirs(), save_patches(), find_tampered_patches(), patch_extractor_driver

Class: PatchExtractorCASIA

Training Accuracy and Loss: Complete

Self-Consistency Learning:

Module Image Preprocessing and feature map generation:
Functions: load_image(), `get_stride_new_dim()`,`get_new_stride()`,`get_patches()`, `reshape_map()`,`get_response_maps()`

Aravind J:

CNN Approach:

SVM Classifier:

Functions: classify(),print_confusion_matrix(),classify(),get_predictions()

Delf-Consistency Learning:

Class: Resnet_backbone,Siamese

Pranay Varma:

CNN Approach:

Module Model Training
Functions :get_filters(),vectorize_filters(),create_loss_and_optimizer(),train_net()

Class:CNN

Self-Consistency Learning:

Functions: Mean_Shift(),Normalized_Cut(),get_meanshift_ncut()

Difficulties faced:

i) **New Tech Stack**: None of our team members have used PyTorch before. So, we all had to learn it from scratch by referring the official docs.

ii) **Computational Power:** We soon realized that our local machines will not be able to handle the training of the neural networks and the processing of the high-resolution images, so we had to switch to Kaggle where we have access to 30 hours of free GPU per week (around 10 GB can be utilized).

Guide Signature