IMAGE FORGERY DETECTION USING CNN

This project focuses on detecting forgeries in digital images using deep learning techniques, specifically Convolutional Neural Networks (CNN). Image forgery detection is a critical task in digital forensics, aiming to identify tampered images that have been altered using editing techniques such as copy-move, splicing, or resampling.

Project Overview: The project leverages a CNN architecture to automatically learn features that distinguish between genuine and forged images. By feeding the network image data, the model learns complex patterns related to tampering artifacts. The architecture includes custom filters applied to the second convolutional layer.

The model is trained on a labelled dataset consisting of both authentic and forged images, and the task is framed as a binary classification problem.

1. Dataset

- **Description of the Dataset**:
  - CASIA 2.0 dataset
  - Number of images 12,614(real and forged)
- **Preprocessing Steps**:
  - Extract patches of size 128x128x3 from image

2. CNN Architecture

- **CNN Model** :
  - Input size : 128x128x3
  - Number of layers : 9

- Output Size of CNN Model : 16x5x5 feature vector extracted from 128x128x3

Layer	Input Size	Output Size	Stride	Padding	Kernel	Activation Function
1st	128x128x3	124x124x30	1	0	5x5	Relu
2nd	124x124x30	30x30x16	2, 2	0	5x5, 2x2	Relu, Maxpool
3rd	30x30x16	28x28x16	1	0	3x3	Relu
4th	28x28x16	26x26x16	1	0	3x3	Relu
5th	26x26x16	24x24x16	1	0	3x3	Relu
6th	24x24x16	11x11x16	1, 2	0	3x3, 2x2	Relu, Maxpool
7th	11x11x16	9x9x16	1	0	3x3	Relu
8th	9x9x16	7x7x16	1	0	3x3	Relu
9th	7x7x16	5x5x16	1	0	3x3	Relu

- **Hyperparameters**:
  - Loss Function : Cross entropy loss (Best for binary classification)
  - Optimizer : Stochastic gradient descent (momentum)
  - Scheduler : Step LR (to decay learning rate by 0.9 for every 10 epochs)
  - Epoch : 50
  - Learning Rate : 0.0001
  - Batch size : 128

- **SVM Model**
  - Input Size : 16x5x5 features
  - Kernel : Radial Basis Function
  - Gamma : 1e^-3, 1e^4
  - C : 0.001, 0.01, 0.1, 1 ,10, 100, 1000
  - Output Size : Forged or Authentic

- **Framework/Tools**:
  - PyTorch

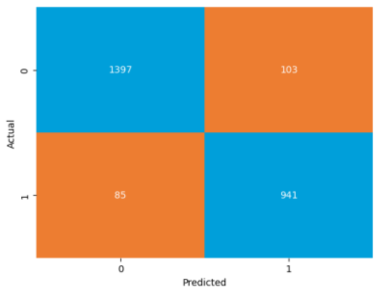
- OpenCV
- Scikit-learn
- NumPy
- Pandas

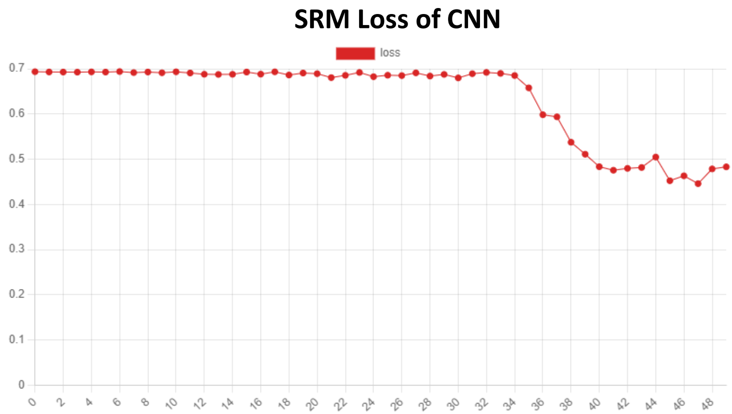
- **Code Structure**: Provide an overview of the key Python files and their roles:
    - Data loading
    - Model definition
    - Patch Extraction
    - Feature Vectorization
    - Feature Fusion
    - Training Model
    - Evaluation

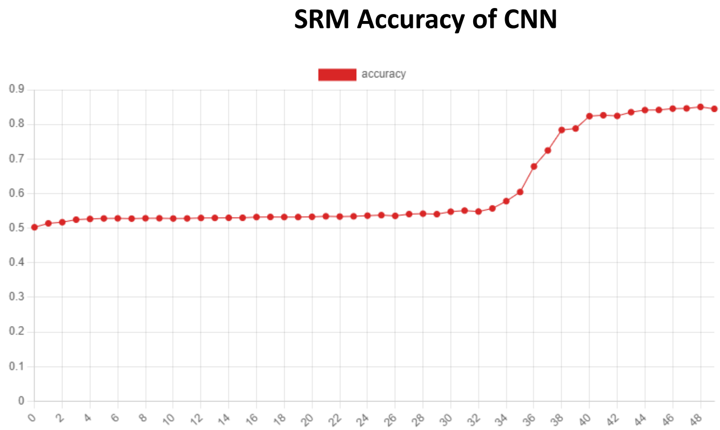
4. Evaluation

- **Performance Metrics**: Metrics used to evaluate the model:
    - Accuracy
    - Loss
- **Visualization:**

SVM Model Confusion Matrix







5. Results

- **Final Accuracy**:
  - CNN Model = **84.4712%**
  - SVM Model = **92.4757%**

6. Challenges and Limitations

- Issues faced during the project, such as:

    - Overfitting : We used single dropout technique to overcome this challenge
    - Computational challenges : Requires high computational power to process the data of the dataset

8. Future Work

- Additional architectures to explore (e.g., GAN-based methods)
- Fine-tuning and hyperparameter optimization

9. Authors

- Modagampalli Mani Datt (1005-21-733036)
- Kspsvln Siddardha Kumar Kavuri (1005-21-733058)