

Date: 12/09/2024 (Fall 2024 Semester)

Final Project Presentation

Image Forgery Detection



COURSE: Data Science with Python

COURSE INSTRUCTOR: Eugene Pinsky

BY SRUJANA NIRANJANKUMAR (BU ID: U61717332)



Table of Contents

1. Background
2. Motivation
3. Problem Statement
4. Project Objectives
5. Previous Related Work
6. Image Forgery Types
7. CNN Architecture
8. Design
9. Implementation
10. Results
11. Performance + Metrics
12. Conclusion



Background

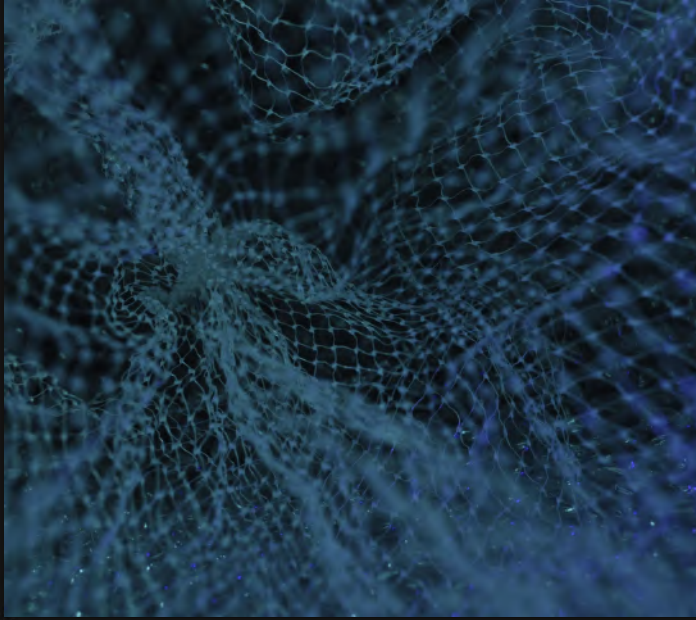
Image forgery detection is the process of identifying images that have been manipulated or altered to deceive or mislead viewers.

Modern digital technology has started to undermine the authenticity of images, enabling forgeries and introducing a darker side to its capabilities.


A variety of software tools have been developed for image processing, which can be used to create convincing forged images that appear real.

In an age of digital manipulation, verifying the authenticity and reliability of images is essential in areas such as journalism, forensics, and law enforcement.


Motivation



The motivation behind image forgery detection using CNNs stems from the increasing frequency of image manipulation and the urgent need for effective methods to address it, as it contributes to the distortion of visual content.



The aim is to create a CNN model capable of differentiating between genuine and manipulated images.



The objective is to safeguard the integrity of digital media, ensure the accuracy of evidence, and uphold trust in visual information.



Ultimately, the goal is to contribute to the development of robust tools that improve the security and reliability of digital images.

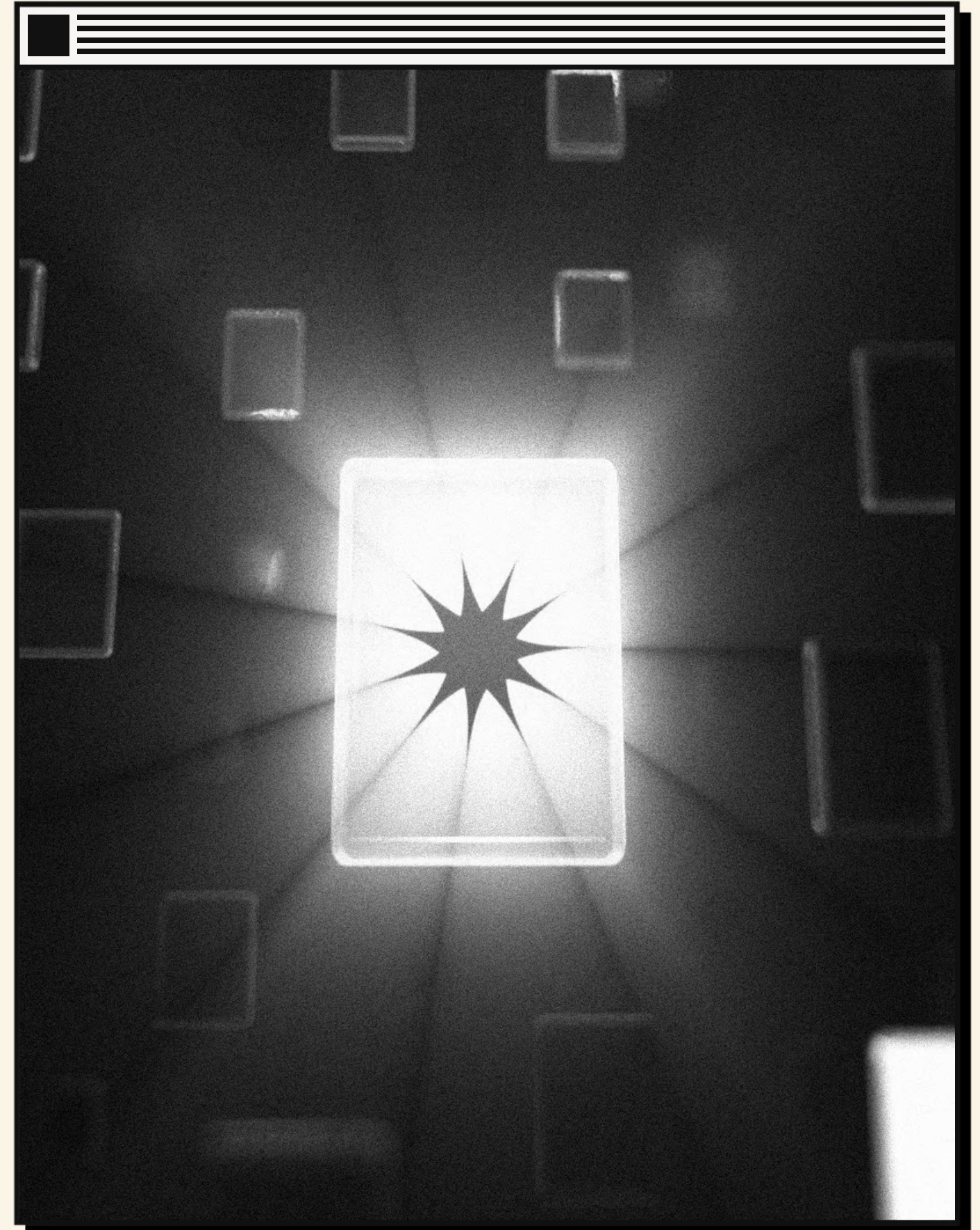
Problem Statement

In today's digital age, image manipulation is widespread, raising concerns about the authenticity of visual content.

This project focuses on detecting and classifying image forgeries—such as copy-move, splicing, and retouching—using Convolutional Neural Networks (CNNs).

The challenge is to develop a robust detection system that can adapt to increasingly sophisticated forgeries while ensuring efficient and accurate results across various scenarios.

Key challenges include evolving forgery techniques, limited diverse training data, and adversarial attacks.





Project Objectives

- Develop a Convolutional Neural Network (CNN) to detect instances of copy-move, splicing, and retouching in digital images.
- Train and refine the CNN model to automatically extract key features that signal image forgery.
- Provide a versatile tool for digital forensics experts, investigators, and researchers, improving their ability to identify and verify the integrity of visual content.

Previous Related Work

There are numerous projects and research studies related to **image forgery detection**, an area that intersects with computer vision, machine learning, and cybersecurity. Here are some notable existing projects and techniques:

Forgery Type	Description	Techniques/Tools	Example Projects
Image Splicing	Detecting parts of an image pasted from another.	Machine learning classifiers, CNNs, edge/color analysis.	CASIA Image Tampering Dataset
Copy-Move Forgery	Identifying duplicated regions within the same image.	Keypoint-based (SIFT, SURF), block-based methods, CNNs.	OpenCV implementations
Deepfake Detection	Detecting AI-generated face or video manipulations.	Temporal/spatial analysis, GAN detectors (MesoNet, XceptionNet).	Kaggle's Deepfake Detection Challenge
Image Retouching	Identifying subtle enhancements like brightness/color adjustments.	Error Level Analysis (ELA), residual noise analysis.	Photo Forensics
Forgery Localization	Pinpointing manipulated regions within an image.	Patch-based CNNs, segmentation models (UNet).	NIST Media Forensics Challenge (MFC)
Blockchain-based	Ensuring authenticity using blockchain to track image history.	Cryptographic hashing, decentralized storage.	Truepic, OriginStamp

Previous Related Work - Research Related to Image Forgery Detection

Image Forgery Detection using Deep Neural Network (2021) - Anushka Singh, Jyotsna Singh

- Explores a passive authentication technique for image forgery detection.
- Implements a basic convolutional neural network (CNN) combined with two preprocessing methods.
- Compares the performance of transfer learning models such as VGG16 and ResNet50.
- Observes that ResNet50 achieves the highest accuracy compared to VGG16 and the standalone CNN.

Image Forgery Detection using Machine Learning (2021) - Shanthraj, Selvaraj, Ramya IM

- Introduces CNN-based approaches for detecting forged images.
- Addresses the limitations of conventional forgery detection techniques.
- Highlights the inadequacy of traditional methods in handling diverse tampering techniques.
- Demonstrates that deep learning can effectively identify complex and abstract features essential for forgery detection.

Identifying Fake Images through CNN-based Classification using FIDAC (2022) - Shraddha Pawar, Bhavin Goswami, Gaurangi Pradhan, Sonali Bhutad

- Utilizes Error Level Analysis (ELA) for preprocessing.
- Employs the VGG model for classification tasks.
- Conducts evaluation using the CASIA dataset and testing with FIDAC, demonstrating improved accuracy when combining these datasets for testing.

Previous Related Work - Research Related to Image Forgery Detection (Continued...)

Detection of Spliced Images in Social Media Applications (2021) - Munera A. Jabaar, Saad N. Alsaad

The detection process is divided into two main phases:

- Preprocessing: Images are prepared using Error Level Analysis (ELA).
- Model building: Includes image normalization and enhancement for CNN-based detection.

Copy-Paste Forgery Detection using Deep Learning with Error Level Analysis (2023) - N. V. S. K. Vijayalakshmi K, J. Sasikala, C. Shanmuganathan

- The methodology consists of three stages: preprocessing, image augmentation, and classification.
- Preprocessing involves image normalization, rescaling, and ELA application.
- The system is tested on the MICC-F220 dataset and implemented using Python.

A Deep Learning Framework for Copy-Move Forgery Detection in Digital Images (2022) - Navneet Kaur, Neeru Jindal, Kulbir Singh

The framework includes three components:

- Feature extraction using a deep learning model (AlexNet).
- Feature matching with a lexicographically sorted matrix for efficiency.
- Post-processing to eliminate potential false matches.

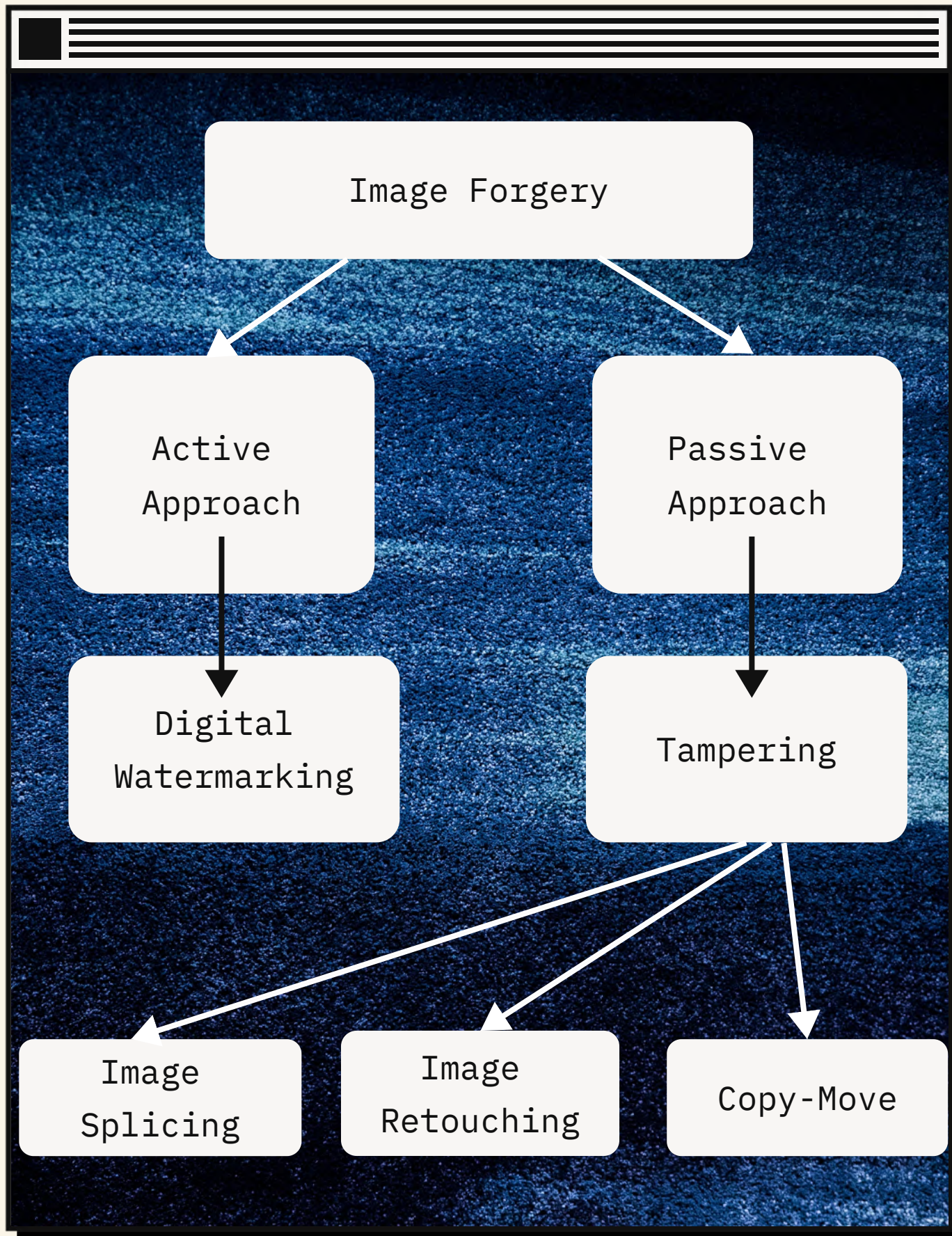


Image Forgery Types

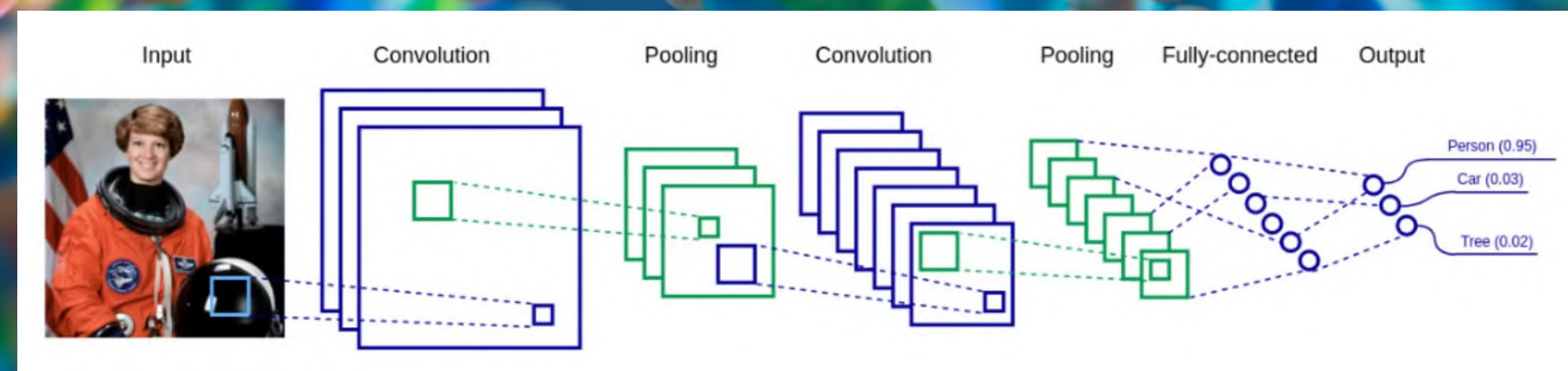
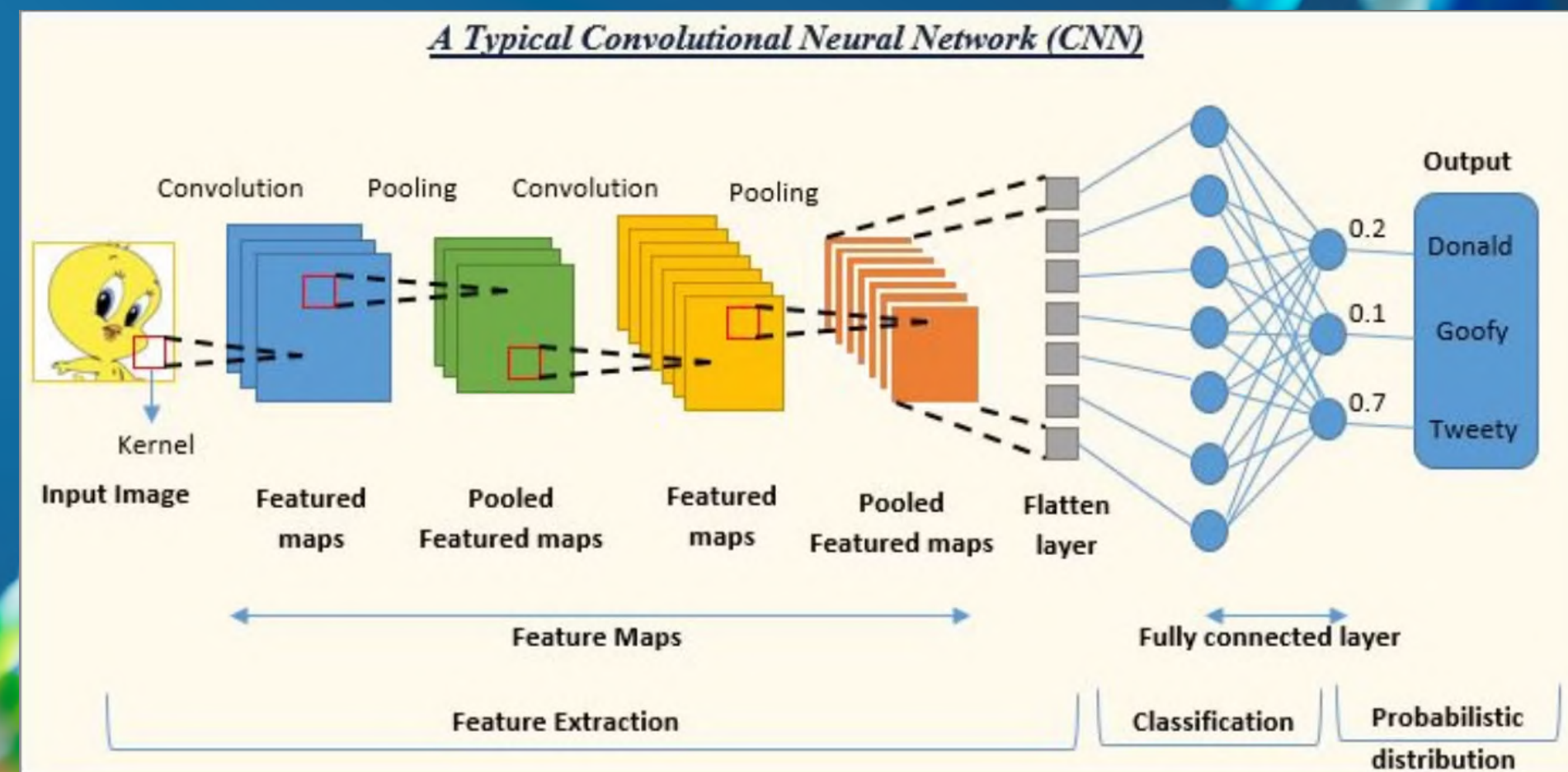
The active approach involves embedding additional information, such as a watermark or digital signature, into the image during its creation or before distribution.

- **Digital watermarking:** Embedding hidden data into digital media, such as images, audio, or video, for identification or verification purposes.

The passive approach, or blind detection, examines the image itself for inconsistencies or anomalies that could suggest tampering.

- **Copy-Move Forgery:** Copying a section of an image and pasting it elsewhere within the same image.
- **Image Splicing:** Combining elements from multiple images to create a deceptive, new composition.
- **Retouching:** Altering an image by selectively enhancing or modifying certain areas to achieve a desired effect.

CNN Architecture (Layers of CNN model)



$$y_{row,col} = \sum_{i=1}^{F_h} \sum_{j=1}^{F_w} x_{row+i-1,col+j-1} \times w_{i,j} + b$$

filter

slice

1	2	3
4	5	6
7	8	9

slice

1	2
3	4

+

1	2
3	4

=

37	

1*1+2*2+4*3+5*4=37

slice

1	2	3
4	5	6
7	8	9

slice

1	2
3	4

+

1	2
3	4

=

37	47

2*1+3*2+5*3+6*4=47

slice

1	2	3
4	5	6
7	8	9

slice

1	2
3	4

+

1	2
3	4

=

37	47
67	

4*1+5*2+7*3+8*4=67

slice

1	2	3
4	5	6
7	8	9

slice

1	2
3	4

+

1	2
3	4

=

37	47
67	77

5*1+6*2+8*3+9*4=77

CNN (Convolutional Neural Network) is a deep learning algorithm designed specifically for image recognition and processing tasks. It consists of several layers, including convolutional layers, pooling layers, and fully connected layers.

CNN Details and Core Concepts (1)

CNN Architecture Details

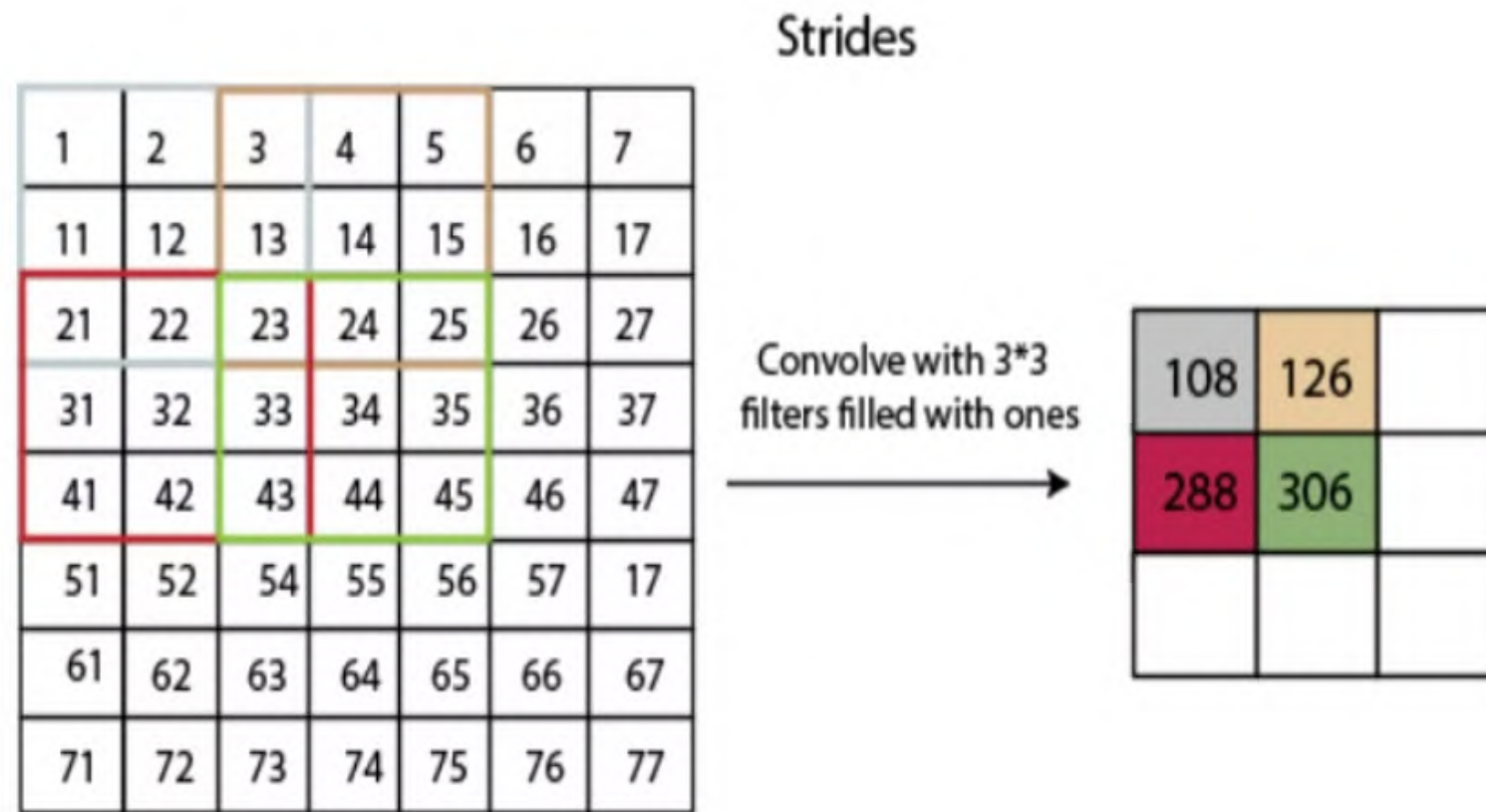
- **Input Layer:** Accepts resized RGB images with a shape of `[None, 50, 50, 3]`.
- **Convolutional Layers:** Comprises five layers with increasing filter sizes (32, 64, 128, 32, and 64), each using ReLU activation for non-linear feature extraction.
- **Pooling Layers:** Max-pooling layers with 3x3 filters are applied after each convolutional layer to reduce dimensionality and retain key features.
- **Fully Connected Layer:** A dense layer with 2048 units, ReLU activation, and a dropout rate of 0.8 to prevent overfitting.
- **Output Layer:** A final layer with 4 units and softmax activation for multi-class classification.

CNN Details and Core Concepts (2)

Core CNN Concepts

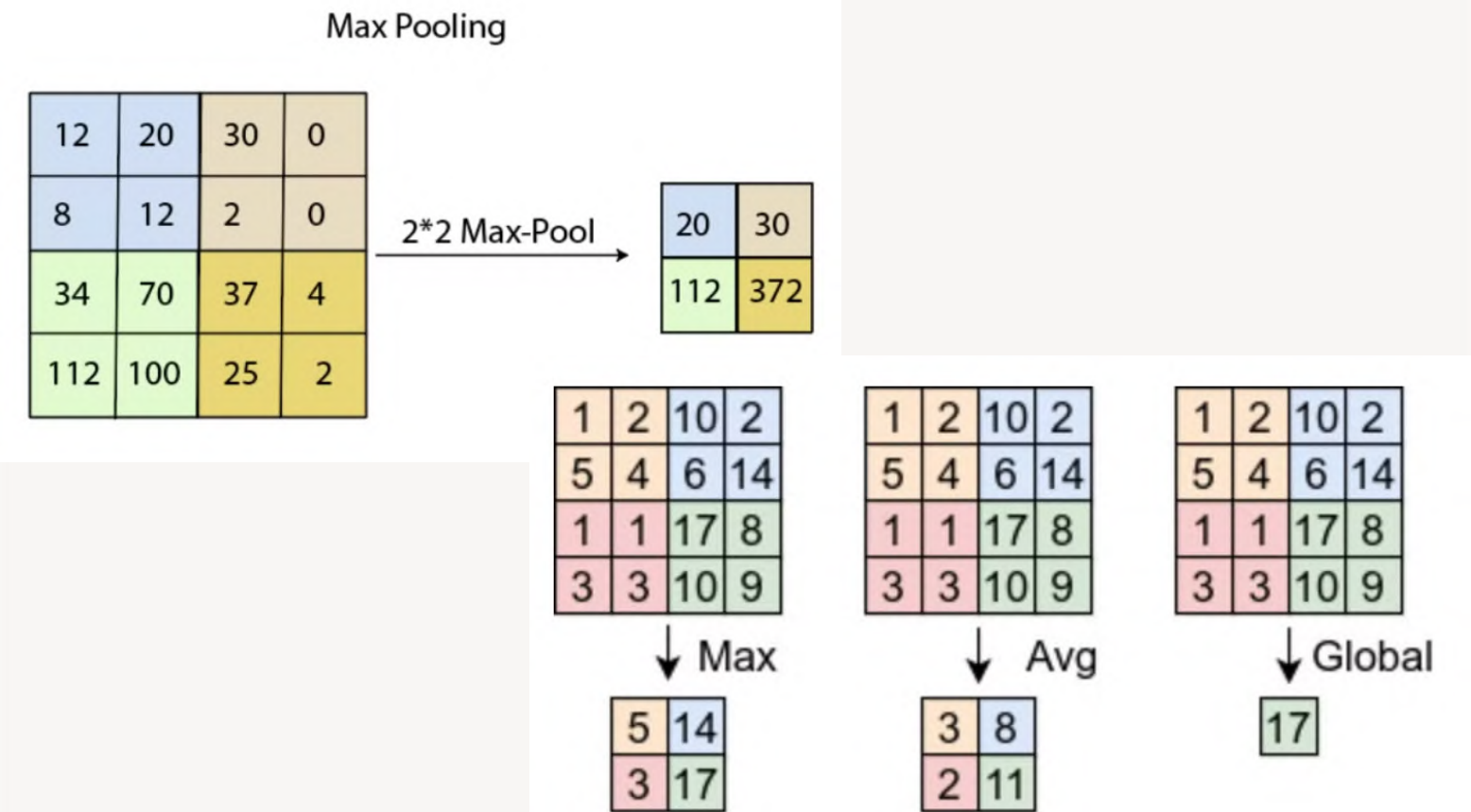
- **Convolutional Layers:** Use filters (e.g., 3x3) to extract key image features by sliding across the image, generating feature maps.
- **Activation Layers:** Introduce non-linearity through ReLU activation, defined as $f(x) = \max(0, x)$, to better capture complex patterns.
- **Pooling Layers:** Perform downsampling to reduce the dimensionality of feature maps while preserving essential information.
- **Fully Connected Layer:** Transforms feature maps into a flattened vector, enabling the model to identify patterns and classify images for effective forgery detection.

Some more details..



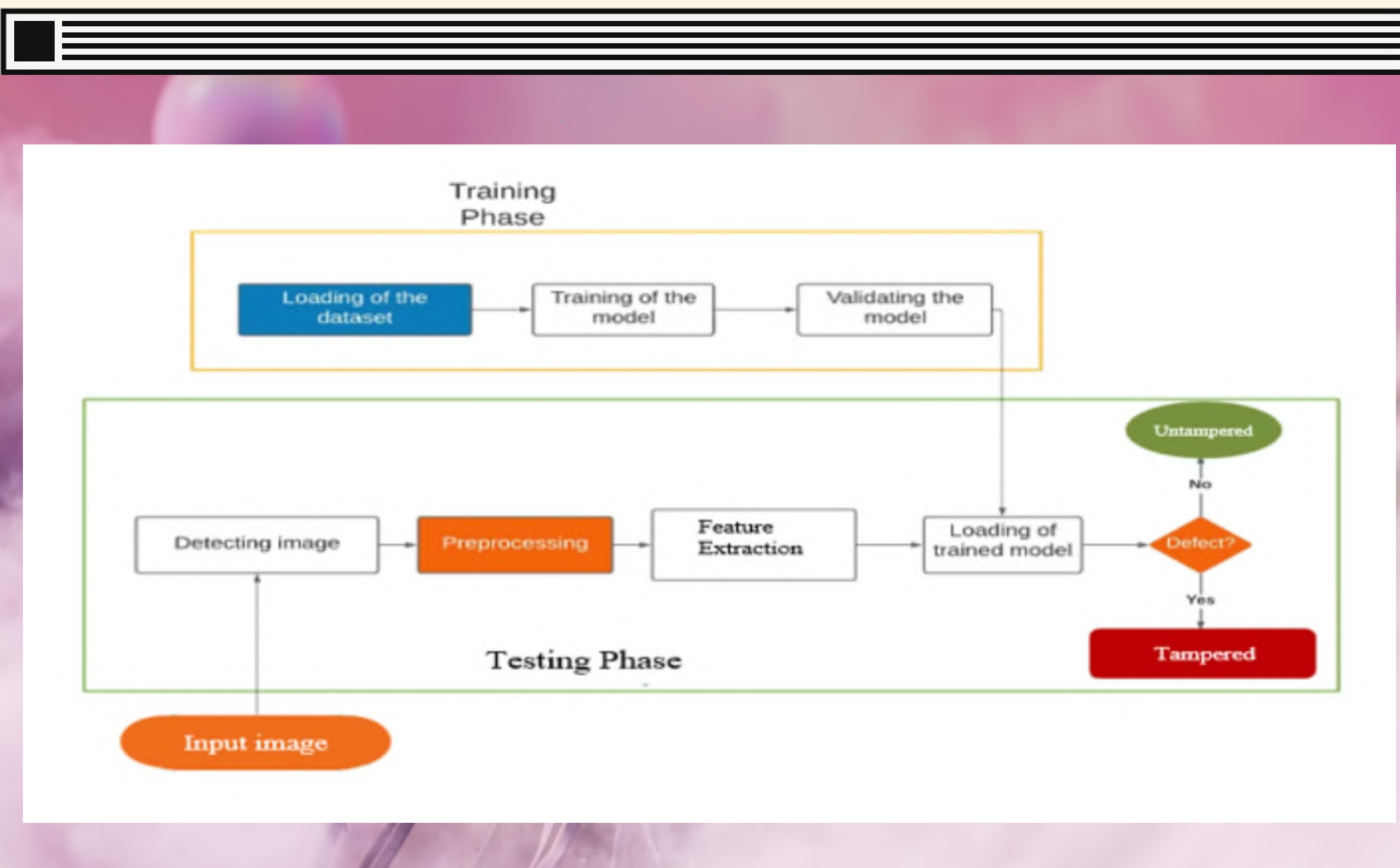
Convolution layer:

This layer is responsible for extracting features from the input dataset.



Pooling layer:

The pooling layer reduces the number of parameters when working with large images by downscaling the image output from the previous layers.

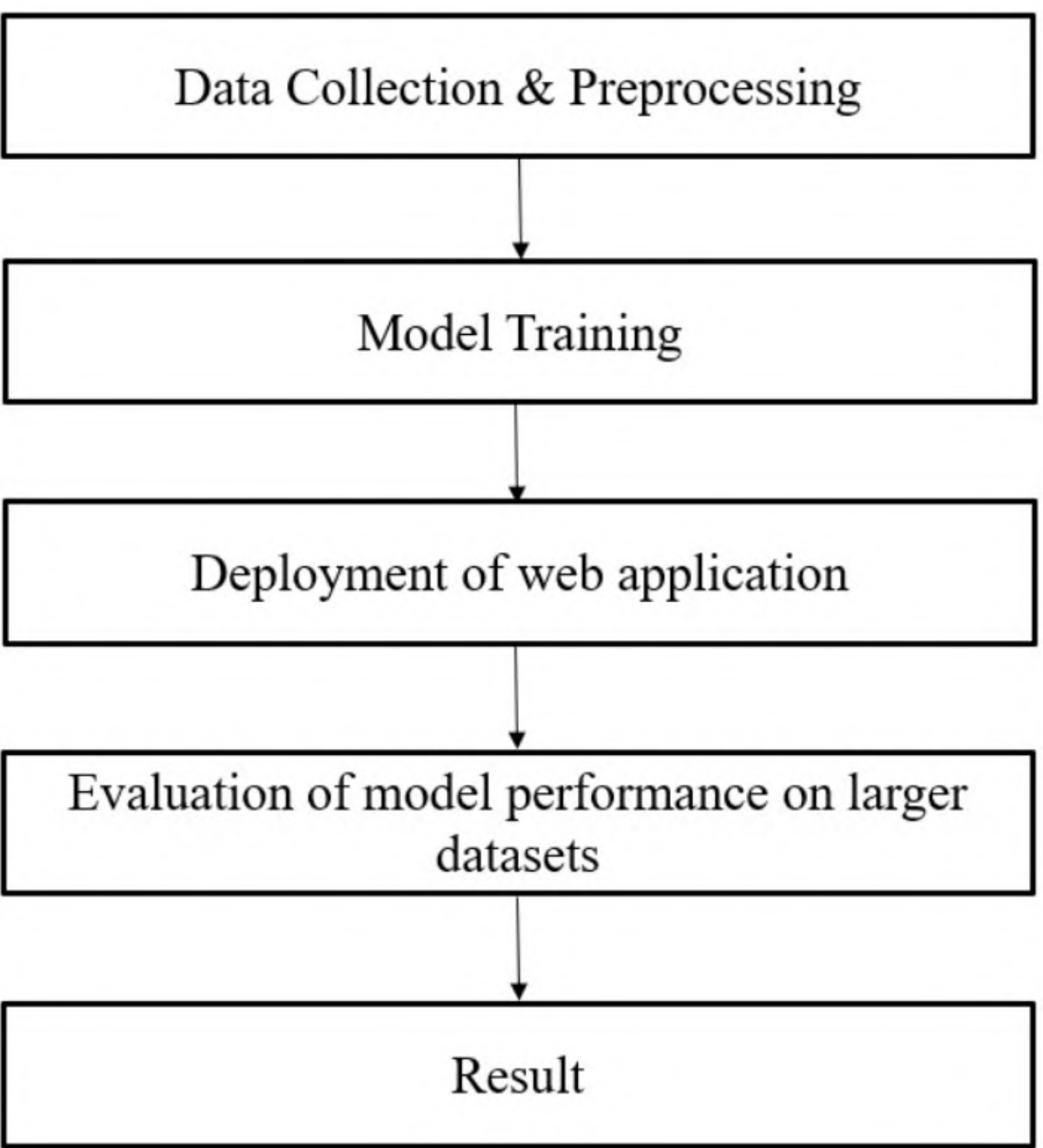


Design

TRAINING PHASE
+
TESTING PHASE

Implementation

- 1.**Dataset Collection:** Gather and compile a diverse dataset that reflects various real-world image manipulation scenarios. I obtained a diverse dataset of genuine and manipulated images from Kaggle, ensuring its reliability and applicability to real-world scenarios.
- 2.**Data Preprocessing:** Apply algorithms to clean and enhance the quality of the dataset images.
- 3.**Model Training:** Train a CNN model using the prepared dataset, fine-tuning it for detecting image forgeries.
- 4.**Model Saving:** Store the trained CNN model in a file for later use.
- 5.**User Image Input:** Enable users to upload an image for analysis.
- 6.**Image Processing:** Preprocess the uploaded image using appropriate techniques.
- 7.**Forgery Prediction:** Use the saved CNN model to analyze the uploaded image and determine if forgery is present.



Data Collection & Preprocessing

Model Training

Deployment of web application

Evaluation of model performance on larger
datasets

Result

Flowchart

This flowchart outlines a project workflow:

- **Data Collection & Preprocessing:** Gather and clean data for analysis.
- **Model Training:** Train a machine learning model using the preprocessed data.
- **Deployment of Web Application:** Deploy the trained model within a web app for usability.
- **Model Performance Evaluation:** Test the model on larger datasets to assess its performance.
- **Result:** Present the final outcomes or conclusions.

This represents an end-to-end ML pipeline integrated with deployment.

Results

Terminal Output

Website is hosted on
`http://127.0.0.1:5000` after
running code in python virtual
environment ——>

Website Image Forgery Analysis Output

Next slide...

```
DataScienceFinalProject — Python < Python app.py — 80x24

Last login: Sun Dec  8 02:16:40 on ttys001
[sru@Srujanas-MacBook-Pro DataScienceFinalProject % python3 -m venv myenv ]
[sru@Srujanas-MacBook-Pro DataScienceFinalProject % source myenv/bin/activate ]
(myenv) [sru@Srujanas-MacBook-Pro DataScienceFinalProject % python3.9 app.py ]
WARNING:tensorflow:From /opt/homebrew/lib/python3.9/site-packages/tensorflow/pyt
hon/compat/v2_compat.py:98: disable_resource_variables (from tensorflow.python.o
ps.resource_variables_toggle) is deprecated and will be removed in a future vers
ion.
Instructions for updating:
non-resource variables are not supported in the long term
* Serving Flask app 'app'
* Debug mode: on
WARNING: This is a development server. Do not use it in a production deployment.
Use a production WSGI server instead.
* Running on http://127.0.0.1:5000
Press CTRL+C to quit
* Restarting with stat
WARNING:tensorflow:From /opt/homebrew/lib/python3.9/site-packages/tensorflow/pyt
hon/compat/v2_compat.py:98: disable_resource_variables (from tensorflow.python.o
ps.resource_variables_toggle) is deprecated and will be removed in a future vers
ion.
Instructions for updating:
non-resource variables are not supported in the long term
* Debugger is active!
```


Choose File

no file selected

Analyse



Status: Original

Probability: The predicted image is original with a probability of 99.3013687133789 %

Choose File

no file selected

Analyse



Status: Copy-Move

Probability: The predicted image is Copy-Move with a probability of 99.33934783935547 %

Choose File

no file selected

Analyse



Status: Splicing

Probability: The predicted image is Splicing with a probability of 99.78276062011719 %

Choose File

no file selected

Analyse



Status: Retouched

Probability: The predicted image is Retouched with a probability of 99.84833526611328 %



Performance + Metrics

- **System Performance:** The CNN-based image forgery detection system demonstrated excellent performance, achieving an accuracy of 94% after 50 epochs. It successfully identified multiple types of forgeries, including copy-move and splicing, with consistent improvements in validation accuracy, reflecting strong generalization capabilities.
- **Dataset and Evaluation Metrics:** The system was evaluated using standard datasets such as CASIA, MICC-F220, and the Columbia Image Splicing Dataset, showcasing its robustness. Performance metrics revealed a precision exceeding 90%, highlighting its effectiveness in accurately detecting forged images.

Conclusion

- This project developed an advanced image forgery detection system utilizing Convolutional Neural Network (CNN) classification techniques.
- Comprehensive experimentation and testing demonstrated high accuracy in identifying diverse types of forgeries, including copy-move, splicing, and retouching.
- The CNN-based approach leverages hierarchical feature extraction, eliminating the reliance on manual feature engineering and improving adaptability and efficiency.
- This innovation addresses a critical societal challenge by providing reliable detection of digital manipulation in an era of widespread image editing.
- The system is a valuable resource for digital forensic analysts, law enforcement agencies, journalists, and industries dependent on trustworthy visual content, promoting authenticity and integrity in digital media.



Thank you!



Student Email: `sruju333@bu.edu`