### **Comparative Analysis of Feature Extraction Techniques for Robust Deepfake Image Detection**

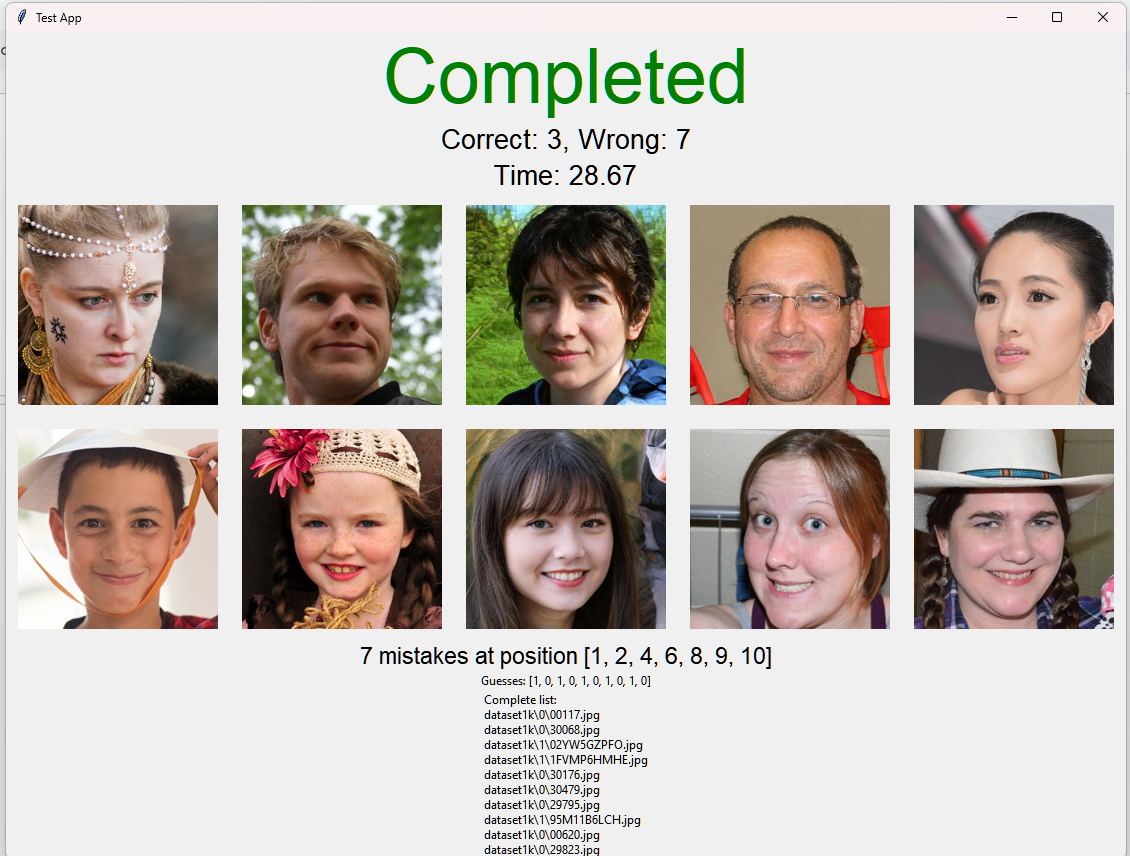
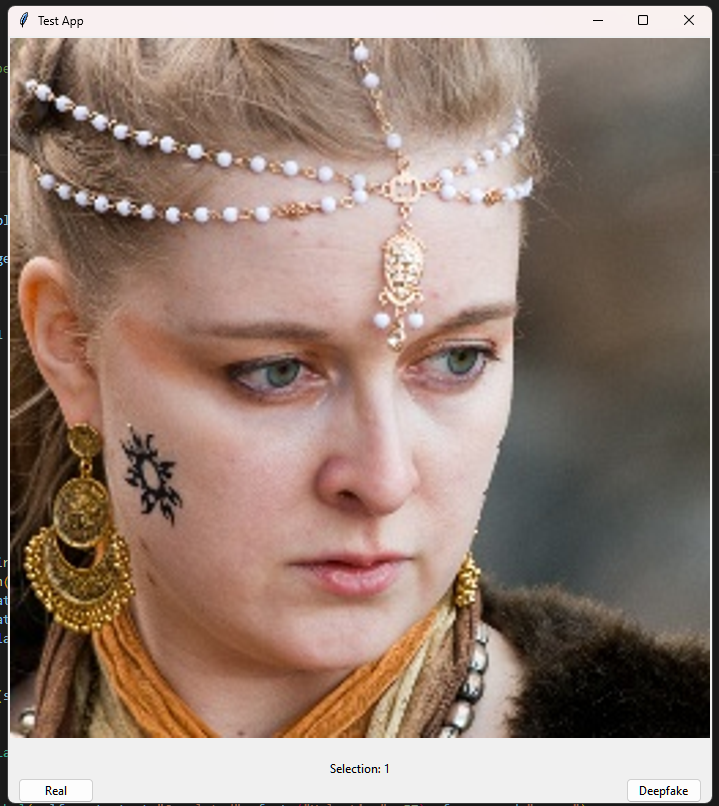
### **Introduction to the Problem**

Deepfake image detection has become a critical challenge in computer vision due to the growing ease of generating highly realistic synthetic media using artificial intelligence. Deepfakes, which involve the generation of human faces in videos or images, pose significant ethical, legal, and security concerns. Detecting such manipulated content demands robust techniques capable of distinguishing between authentic and generated images.

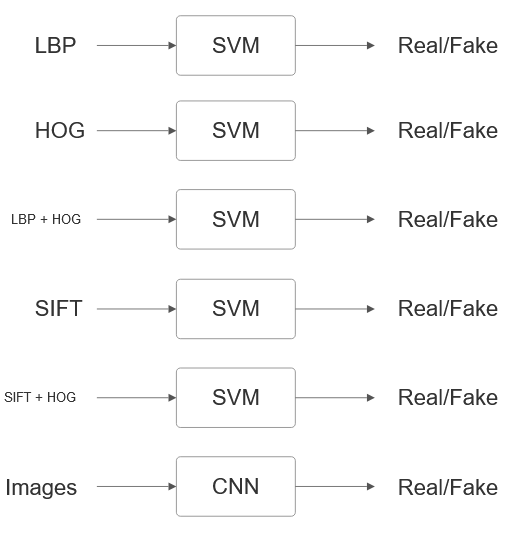
This project aims to investigate and compare various feature extraction techniques for deepfake image detection. By evaluating the performance of traditional feature extraction methods, such as *Local Binary Patterns* (LBP) and *Histogram of Oriented Gradients* (HOG), alongside a neural network-based end-to-end model, the project tries to identify the strengths and weaknesses of each approach.

**Human Performance**  
  
Through an app developed for this project, 24 participants were tested using the same dataset employed for training and evaluation. Each participant was presented with 10 images per test, and the time required to complete each test was recorded.

Out of 132 total runs, the mean accuracy was 6.59 correct answers per test, while the average time required was 49 seconds per test.



**Employed Features Extractor**



#### **1. Local Binary Patterns (LBP)**

LBP is a texture descriptor that captures local features by encoding the relationship between a pixel and its neighboring pixels.

### **Advantages:**

* **Captures fine-grained Textures:** LBP could be effective in detecting irregular patterns or inconsistencies in skin texture due to its focus in local texture.
* **Computational Efficiency:** LBP is computationally lightweight, making it well-suited for analyzing large datasets.
* **Baseline for Comparison:** due to its low computational cost, it will be used as a baseline for comparison.

### **Limitations:**

* Struggles with high-level features, such as global structures, limiting its ability to capture complex anomalies.

#### **2. Histogram of Gradients (HOG)**

HOG focuses on capturing the gradient orientation distributions of an image, making it effective in detecting shapes and edges.

### **Advantages:**

* **Edge and Shape information:** Highlights distortions and artifacts in edges and contours.
* **Complementary to LBP:** Combining HOG and LBP can provide a more comprehensive image representation.

### **Limitations:**

* Less effective at capturing fine-grained texture details.

#### 

#### **3. Scale-Invariant Feature Transform (SIFT)**

SIFT identifies and describes local keypoints in an image.

This project explores two SIFT implementations:

1. **Dense SIFT**: Focuses on predetermined points arranged on a grid, providing a uniform analysis of the image.

2. **Mean SIFT Descriptors**: Aggregates the mean of all SIFT descriptors found in a given image, offering a compact representation.

### **Advantages:**

* **Focus on keypoints:** Analizes specific regions of the image.
* **Robustness:** Performs well under transformations like scaling and rotation.
* Relatively compact (Mean SIFT)

### **Limitations:**

* Computationally intensive compared to simpler methods.
* The descriptors generated by SIFT are high-dimensional (128 per key point), leading to large feature vectors if many key points are detected (Dense SIFT)

For all the previously mentioned features extractor, classification will be performed by SVM model trained for each specific feature extractor (or combination of feature extractor).

#### **4. Convolutional Neural Network (CNN)**

A deep neural network is designed to perform feature extraction and classification in a single pipeline. This end-to-end model is trained directly on raw image data, enabling it to learn hierarchical features through multiple convolutional layers.

### **Advantages:**

* **Feature Extraction:** Automatically extracts features at multiple levels, from low-level textures to high-level patterns.
* **Integrated Workflow:** Combines feature extraction and classification.

### **Limitations:**

* Requires significant computational power and large datasets for effective training.
* Functions as a 'black box,' making it difficult to explain decisions.

### **Model architecture:**

**Dataset**

The dataset for this project includes both real and fake images:

* Real images: Samples were sourced from Flickr-Faces-HQ (FFHQ), provided by Nvidia.
* Fake images: Faces were sampled by the “1 Million Face Faces” dataset, created with StyleGan and provided by Bojan Tunguz.

**Training data**

* A total of 3000 samples were used for training, evenly split between 1500 real images and 1500 fake images.
* For simpler feature extraction methods (such as LBP), an extended dataset of 20.000 total samples was used; however, it led to similar performance while significantly increasing training time.

**Evaluation Metrics**

The performance of the feature extraction methods and classification models will be evaluated using the following criteria:

1. **Accuracy**: Evaluated using the F1-score to measure the balance between precision and recall in the classification.
2. **Computational Efficiency**: Feature extraction times and training times will be compared for each method to assess computational performance.
3. **Robustness to Adversarial Attacks**: Using adversarial attack generation techniques (Adversarial Robustness Toolbox), the resilience of each model against perturbed inputs will be analysed. In particular, adversarial samples have been generated with Fast Gradient Method (FGM) and Carlini and Wagner L2 Method (CL2).

**Results**

|  |  |  |  |
| --- | --- | --- | --- |
| FEATURE EXTRACTOR | Training time (seconds) | accuracy | f1-score |
|  |  |  |  |
| **LBP** | **86,17** | **0,64** | **0,64** |
| **HOG** | **67,97** | **0,74** | **0,74** |
| **LPB+HOG** | **1908,29** | **0,82** | **0,82** |
| **SIFT** | **47,84** | **0,62** | **0,62** |
| **Dense SIFT** | **1848,04** | **0,82** | **0,82** |
| **Mean SIFT + HOG** | **2125,73** | **0,81** | **0,81** |
| **CNN (20ep)** | **919,38** | **0,69** | **0,69** |
| **CNN (10ep)** | **456,57** | **0,7** | **0,7** |
| **CNN (5ep)** | **232,03** | **0,67** | **0,66** |

|  |  |  |  |
| --- | --- | --- | --- |
| FEATURE EXTRACTOR | Training time (%) | accuracy (%) | f1-score (%) |
|  |  |  |  |
| **LBP** | **100** | **100** | **100** |
| **HOG** | **78,88** | **115,62** | **115,62** |
| **LPB+HOG** | **2214,56** | **128,12** | **128,12** |
| **SIFT** | **55,52** | **96,88** | **96,88** |
| **SIFT+HOG** | **2466,9** | **126,56** | **126,56** |
| **CNN (20ep)** | **1066,94** | **107,81** | **107,81** |
| **CNN (10ep)** | **529,85** | **109,38** | **109,38** |
| **CNN (5ep)** | **269,27** | **104,69** | **103,12** |

Training time:

* Mean SIFT has the shortest training time (47,84 seconds), followed by HOG (67,97 seconds) and LBP (86,17 seconds), showcasing their computational efficiency. CNN (with 5 epochs) has a reasonable training time (232.03 seconds)
* LBP + HOG (1908.29 seconds) and SIFT + HOG (2125.73 seconds) and Dense SIFT require significantly longer training time due to the concatenation of features or the high number of keypoints, resulting in high-dimensional feature vectors. CNN ( with 20 epochs) required 919.38 seconds, which is a substantial increase compared to CNN with fewer epochs.

Accuracy and F1-score

* Dense SIFT (0,82), LBP + HOG (0.82), SIFT + HOG (0.81) were the best performers, achieving the highest accuracy and f1-score, demonstrating the benefit of combining complementary features (texture from LBP and edges from HOG)
* HOG (0.74) achieved a strong performance despite its simplicity and fast training time, outperforming LBP (0,64) individually.This shows that edge-based global features are more informative than localized texture or key-point-based ones.
* CNN (10 epochs) achieves an accuracy of 0.70, slightly outperforming its counterpart with 5 epochs and approaching HOG’s performance
* Mean SIFT and LBP have the lowest accuracy, showing that these features extractor may lack the discriminative power needed for the classification task.

HOG offers an excellent balance between training time and accuracy. Best performances were delivered by LBP + HOG, SIFT + HOG and Dense SIFT but at a significant computational cost.  
With more training and architectural optimizations, CNNs could achieve competitive or superior performance while offering the advantage of end-to-end learning.

**Adversarial Robustness**

Adversarial robustness was evaluated using two attacks from the Adversarial Robustness Toolbox (ART):

1. **Fast Gradient Method (FGM)**An extension of the Fast Gradient Sign Method, FGM computes the gradient of the loss function with respect to the input features. Adversarial examples are generated by perturbing the input features in the direction that maximizes the loss function. This is a straightforward and computationally efficient attack method.
2. **Carlini and Wagner L2 Attack (CL2)**A state-of-the-art iterative method, CL2 formulates adversarial example generation as an optimization problem. It seeks the minimal perturbation required to misclassify the model, offering a more precise and effective attack compared to FGM.

**LBP**

| LBP | Mean of original L2 norms | Mean of L2 norm of differences | % wrt mean of original L2 norms | Successful attacks (out of 100) |
| --- | --- | --- | --- | --- |
|  |  |  |  |  |
| FGM eps = 0.01 | 4,7768 | 0,051 | 1,067 | 5 |
| FGM eps = 0.03 | 4,7768 | 0,153 | 3,2 | 15 |
| FGM eps = 0.08 | 4,7768 | 0,4079 | 8,53 | 49 |
| CL2 c = 0.05 | 4,7768 | 0,5024 | 10,51 | 95 |
| CL2 c = 0.08 | 4,7768 | 0,5491 | 11,49 | 95 |
| CL2 c = 0.1 | 4,7768 | 0,5695 | 11,92 | 95 |

* FGM Attacks:
  + LBP demonstrates relative robustness against FGM, with only 49% of successfull attacks even at high epsilon value (0.08)
  + For ϵ=0.08, the perturbation results in an 8.53% variation in the L2 norms: adversarial examples cause moderate distortion.
* CL2 Attacks:
  + LBP is vulnerable to the more advanced Carlini and Wagner L2 attack. Even at the lowest confidence (c = 0.05). the success rate is 95%
  + The percentage variation in the L2 norm is on pair to FGM (at ϵ=0.08), yet the success rate of CL2 is substantially higher, indicating that LBP fails under iterative and optimized attacks

Despite its simplicity and computational efficiency, LBP features offer moderate robustness against straightforward attacks like FGM. However, they are highly susceptible to sophisticated attacks such as CL2.

**HOG**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| HOG | Mean of original L2 norms | Mean of L2 norm of differences | % wrt mean of original L2 norms | Successful attacks (out of 100) |
|  |  |  |  |  |
| FGM eps = 0.01 | 89,7785 | 0,9 | 1,002 | 22 |
| FGM eps = 0.03 | 89,7785 | 2,7 | 3,007 | 73 |
| FGM eps = 0.08 | 89,7785 | 7,2 | 8,019 | 100 |
| CL2 c = 0.05 | 89,7785 | 0,3745 | 0,4171 | 41 |
| CL2 c = 0.08 | 89,7785 | 0,3963 | 0,4414 | 40 |
| CL2 c = 0.1 | 89,7785 | 0,3464 | 0,3858 | 36 |

* FGM Attacks:
  + HOG features show vulnerability to FGM attacks, At lower values of epsilon, 22% of the attacks were successfull despite only a 1% difference in the L2 norms.
  + At higher values of epsilon, the attack success rate reaches 100%.
* CL2 Attacks:
  + HOG performs relatively well against Carlini and Wagner L2 attacks. For all tested confidence, the percentage difference in norms remained under 0.5%, leading to low success rate (36-41%).

HOG performs poorly under straightforward attacks like FGM, especially with higher epsilon values, where the success rate rapidly escalates to 100%. However, it exhibits greater robustness to advanced attacks like CL2, achieving relatively low success rates (36–41%).

**LBP + HOG**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| LBP + HOG | Mean of original L2 norms | Mean of L2 norm of differences | % wrt mean of original L2 norms | Successful attacks (out of 100) |
|  |  |  |  |  |
| FGM eps = 0.01 | 162,3967 | 1,6208 | 0,998 | 24 |
| FGM eps = 0.03 | 162,3967 | 4,8624 | 2,9941 | 77 |
| FGM eps = 0.08 | 162,3967 | 12,9664 | 7,9843 | 99 |

* FGM Attacks:
  + The concatenated LBP + HOG features performs similarly to HOG alone for FGM attacks. At ϵ=0.01, close to one fourth of the attacks succeeded, while increasing ϵ to 0.08 yields a success rate of 99%
* CL2 Attacks:
  + Due to the high dimensionality of the concatenated features and the iterative nature of the Carlini and Wagner L2 attack, computation for this method was not feasible with the available resources.

The combination of LBP and HOG features offers no significant improvement in robustness against FGM attacks, and computational limitations prevent the evaluation of performance against CL2 attacks.

**Mean SIFT**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Mean SIFT | Mean of original L2 norms | Mean of L2 norm of differences | % wrt mean of original L2 norms | Successful attacks (out of 100) |
|  |  |  |  |  |
| FGM eps = 0.01 | 11,0991 | 0,1131 | 1,019 | 16 |
| FGM eps = 0.03 | 11,0991 | 0,3394 | 3,057 | 45 |
| FGM eps = 0.08 | 11,0991 | 0,9051 | 8,1547 | 89 |
| CL2 c = 0.05 | 11,0991 | 0,5006 | 4,5102 | 95 |
| CL2 c = 0.08 | 11,0991 | 0,5656 | 5,0959 | 95 |
| CL2 c = 0.1 | 11,0991 | 0,6084 | 5,4815 | 95 |

* FGM Attacks:
  + Mean SIFT shows slightly better performance than HOG for FGM attacks. At ϵ=0.01, only 16% of the attacks succeeded, while at ϵ=0.08 the success rate remains below 90%
* CL2 Attacks:
  + Mean SIFT demonstrates limited robustness to Carlini and Wagner L2 attacks. For all tested confidence levels, the success rate remains consistently 95%.

Mean SIFT offers some resistance to simpler attacks, but it struggles against more sophisticated techniques

**Dense SIFT**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Dense SIFT | Mean of original L2 norms | Mean of L2 norm of differences | % wrt mean of original L2 norms | Successful attacks (out of 100) |
|  |  |  |  |  |
| FGM eps = 0.01 | 313,5703 | 3,1397 | 1 | 30 |
| FGM eps = 0.03 | 313,5703 | 9,4192 | 3 | 81 |
| FGM eps = 0.08 | 313,5703 | 25,1178 | 8.01 | 99 |

* FGM Attacks:
  + Dense SIFT shows worse performance compared to Mean SIFT, possibly due to higher number of input components. At ϵ=0.01 the number of successful attack is 30, but it increases to 81 by increasing epsilon to 0.03. At ϵ=0.08, almost all attacks were successful
* CL2 Attacks:
  + The evaluation of Dense SIFT under CL2 attacks has not been conducted due to computational limitations.

Dense SIFT behaved slightly worse compared to LBP + HOG, despite similar training time required.

**Mean SIFT + HOG**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Mean SIFT + HOG | Mean of original L2 norms | Mean of L2 norm of differences | % wrt mean of original L2 norms | Successful attacks (out of 100) |
|  |  |  |  |  |
| FGM eps = 0.01 | 162,7266 | 1,6239 | 0,9979 | 20 |
| FGM eps = 0.03 | 162,7266 | 4,8718 | 2,9938 | 76 |
| FGM eps = 0.08 | 162,7266 | 12,9916 | 7,9836 | 98 |

* FGM Attacks:
  + The concatenation of Mean SIFT and HOG features performs similar to HOG by itself and worse than Mean SIFT. For ϵ=0.01, only 20% of attacks were successful, and at ϵ=0.08, the success rate, though high, was 98%, marginally better than HOG alone.
* CL2 Attacks:
  + The provided results do not include CL2 evaluations for this combination due to the limited resources available.

Mean SIFT + HOG demonstrates improved robustness against FGM attacks compared to individual methods or other concatenations. However, its effectiveness remains limited at higher ϵ values.

**CNN ( 10 epochs)**

|  |  |  |  |
| --- | --- | --- | --- |
| CNN (10 epochs) | Accuracy on benign samples | Accuracy on adversarial samples | Success rate of attack (%) |
|  |  |  |  |
| FGM eps = 0.01 | 0,7 | 0,59 | 41 |
| FGM eps = 0.03 | 0,74 | 0,54 | 58 |
| FGM eps = 0.08 | 0,71 | 0,54 | 58 |

* FGM Attacks:
  + The CNN demonstrates consistent robustness against FGM attacks across different epsilon values (ϵ=0.01,0.03,0.08). The success rate of attacks remains close to 50% regardless of the increasing epsilon.
  + Even at higher values of epsilon, the model does not show a significant accuracy drop compared to lower epsilon values, suggesting resilience to strong perturbations.
* CL2 Attacks:
  + The Adversarial Robustness Toolbox currently lacks an implementation of the Carlini and Wagner L2 attack specifically for deep learning models.

The CNN trained for 10 epochs exhibits consistent and robust performance against FGM attacks. Its success rate of adversarial attacks remains below 60%, even at high ϵ values, distinguishing it from traditional feature-based methods. This highlights the potential of neural networks to resist gradient-based attacks while maintaining good classification accuracy on benign and adversarial samples.

**Conclusion**

* In this report, the best performance in terms of accuracy and F1-score was achieved by Dense SIFT or by concatenating traditional complementary features, such as LBP and HOG. However they generate features with a high number of components, increasing significantly computational time (~2000% of LBP’s training time) and resulting in models highly vulnerable to adversarial attacks.
* Among traditional methods, the HOG-based model stood out for its balance between performance and computational efficiency. It achieved better accuracy than the tested participants while requiring only ~68 seconds of training time (78.8% of LBP’s training time). In terms of adversarial robustness, it performed comparably to the LBP + HOG model for simpler attacks and demonstrated relatively strong resilience against more complex attacks, with a success rate below 45% at the tested confidence levels.
* CNN models, as shown in prior research, have the potential to achieve higher accuracy levels given sufficient computational resources and larger datasets. However, under the constraints of this study, CNNs underperformed compared to more complex traditional features extractor . Despite their lower accuracy, CNNs demonstrated greater robustness to FGM attacks, maintaining good performance even at higher ϵ values.