NATIONAL UNIVERSITY OF SINGAPORE
DSA4263: Sense-making Case Analysis: Business and Commerce

Final Project (Group 8)
AY 2023/24 Semester 2

Submitted by:
Benedict Halim: A0216056L
Gupta Ananya Vikas: A0226576W
Javier Tham Jun Long: A0216040Y
Peh Fang Ling: A0221977W
Tan Yu Xian: A0222956Y

Word Count: 4934 (Excluding Appendix)

1. Abstract

Recent developments in online dating have led to an increase in online romance scams, posing significant financial and emotional risks to victims. Conventional measures such as legal intervention and awareness campaigns have proven only partially effective. Consequently, there is a need for a technical solution to mitigate these risks. This paper explores the development of a comprehensive model designed to identify profiles by analyzing both image and textual data. We constructed a dataset comprising AI-generated images, ordinary human images, and scam and non-scam profiles. Our model demonstrates promising results, achieving an accuracy of 0.964, with a false positive rate of 0.029361. However, further research is necessary to enhance the model's reliability before it can be deployed as a practical tool for end-users.

2. Introduction

Online romance scams have become a major cybersecurity threat on digital dating platforms worldwide with experts estimating 10% of all profiles in the world to be fake (Strandell,2024). In 2022, nearly 70,000 Americans fell victim to these scams, incurring losses of approximately USD 1.03 billion—more than double the financial impact recorded in 2019 (Federal Trade Commission, 2024). Similarly, Singapore has seen a dramatic rise in scam-related financial losses, escalating from SGD 5.8 million in 2013 to over SGD 30 million in 2022 (Chua, 2023). As digital dating becomes increasingly prevalent, with one in four Singaporeans actively engaging in online dating (Tan, 2024), the urgency for effective preventive measures intensifies.

How do love scams work?

Online romance scams typically unfold over several months, where fraudsters, using dating platforms or social media, develop relationships with victims under the guise of false identities. These identities are often supported by stolen or AI-generated images sourced from the internet. After establishing a connection, scammers usually request money for various fabricated reasons, such as travel expenses to meet the victim. These transactions often involve untraceable payment methods, and invariably, the scammer never fulfils their promises.

3. Research Objective

The aim of this study is to develop methodologies to prevent individuals from falling victim to online dating scams, using a dual approach involving text analysis and image verification:

1. Profile Analysis: We will design a machine learning classifier to differentiate between genuine and fraudulent textual profiles on dating platforms. This involves:
    1.1. Dataset Creation: Compiling a dataset of scammers and genuine profiles.
    1.2. Model Development: Designing a text-based model to assess the authenticity of profile descriptions.
2. Image Verification: We will employ computer vision techniques to distinguish between real and AI-generated profile images. This involves:

2.1. Dataset of Images: Gathering a dataset of AI-generated and real images used in these profiles.
2.2. Image-Analysis Model: Creating a model to verify the authenticity of profile pictures.

3. Combined Evaluation: Testing the effectiveness of each model independently before combining them to enhance scam detection capabilities.

By addressing the issue through both textual and visual data, this project aims to develop a comprehensive solution to identify and mitigate the risks of online romance scams effectively.

4. Data

4.1 Data Sources

Profiles:

- Source Overview: Data for scam and genuine profiles were collected from two distinct platforms. ScamDigger is known for its database of identified scammers, accessible at (http://scamdigger.com/). DatingnMore, available at (http://datingnmore.com/), is recognized for its stringent screening processes aimed at maintaining a scam-free environment.
- Data Types Collected: The dataset includes various elements such as user descriptions, interests, and additional metadata. This rich data collection enables a detailed analysis of prevalent scamming patterns.

Images:

- Source Overview: Real images were sourced from the Flickr FacesHQ dataset, which includes approximately 70,000 images of human faces under different conditions, while AI-generated images were taken from the "1 Million Fake Faces" database. A subset of 10,000 images was selected from each dataset to match the diversity and complexity of one another.
- Image Characteristics: The datasets (Appendix A) include images of varying age, ethnicity, and conditions to ensure a robust classification system.

4.2 Dataset Characteristics

Profiles:

- Dataset Size and Labeling: The final dataset comprises 5,969 profiles, evenly split between scam and genuine categories. This balanced composition facilitates a fair comparison and testing of machine learning models.

Images:

- Dataset Size: The image dataset includes 20,000 images, comprising 10,000 real and 10,000 AI-generated faces. Each image is labeled as fake or real to facilitate the accurate training and validation of the image analysis model.

5. Methods

5.1 Profiles

5.1.1 Data Preparation

- Anonymization and Ethical Compliance:
  - To uphold ethical standards, personally identifiable information (PII) was removed from all genuine profiles to ensure privacy protection. This process included stripping out names, contact information, emails, and real profile pictures.
- Standardization of Profile Data:
  - The dataset was standardized by eliminating features that did not overlap and removing incomplete records to ensure consistency in our analysis. Due to the user-generated nature of the data, fields such as "occupation," "status," and "location" displayed high variability and errors.
  - Discrepancies like varied abbreviations for countries were normalized using Python's Pandas library and regular expressions, ensuring uniformity across the dataset.
- Translation for Uniformity:
  - Non-English profile descriptions were translated into English using the Google Translate API. This step was crucial to avoid biases in a model trained primarily on English data, ensuring uniformity and enhancing the efficacy of subsequent analyses.
- Data Segmentation:
  - The refined dataset consisted of 5,969 profiles, balanced between scam and genuine entries. We divided the dataset into 80% for training and validation and 20% for testing, setting a solid foundation for thorough model training and evaluation.

5.1.2 Feature Engineering

In the feature engineering phase, our dataset initially comprised a raw dataframe that required processing to make it suitable for machine learning algorithms. This involved several feature engineering steps:

1. Text Normalization

- Stemming: Utilizing the Natural Language Toolkit (NLTK), we applied stemming to reduce words to their base forms by trimming affixes. This process simplifies the data, though it may not yield lexically accurate results.
- Stop Word Removal: We removed semantically weak words using NLTK's list of stop words, enhancing the model's focus on significant terms, thereby boosting accuracy and efficiency.

2. Feature Selection
    - Bag-of-Words (BOW): We implemented a unigram model to capture the frequency of word occurrences, ignoring word order but effectively identifying present terms. Various vector sizes were tested (50, 100, 500, 1000, and all words) to find an optimal balance between expressiveness and computational demand.
    - TF-IDF (Term Frequency-Inverse Document Frequency): This method was employed to emphasize words that are critical for accurate classification by evaluating the relative frequency of words in specific documents against the corpus at large.
3. Advanced Feature Engineering Techniques
    - Tukey Method: We applied the Tukey method, using logistic regression outputs to identify significant features based on the interquartile range (IQR). This method, adjusted for different 'k' values (1 to 25), pinpointed features that significantly influence model predictions.
    - Sentence Embedding: Sentence embeddings were used to convert textual descriptions into fixed-size vector representations that preserve semantic meanings and contextual nuances. This technique proved valuable for clustering profiles, aiding in document categorization, topic modeling, and other information retrieval tasks.

5.1.3 Model Selection

To determine the most effective model for classifying profiles as real or scam, we tested a range of algorithms, each selected for its specific strengths in handling textual data and binary classification tasks:

- Naive Bayes (NB): Known for its simplicity and efficiency, Naive Bayes employs Bayes' theorem with the assumption of independence between predictors. It is particularly effective in text classification and remains robust against irrelevant features.
- Logistic Regression (LR): This binary classification algorithm estimates the probability of a profile being a scam by fitting data to a logistic curve. It is preferred for its speed and simplicity, although it necessitates careful management of outliers and feature scaling.
- Support Vector Machine (SVM): SVM excels in identifying the optimal hyperplane that maximizes the margin between two classes. It is versatile, functions well in high-dimensional spaces and adaptable to both linear and nonlinear data through the use of kernel functions.
- Random Forest (RF): As an ensemble method, Random Forest uses multiple decision trees to enhance predictive accuracy and manage overfitting. It is advantageous for its capacity to rank the significance of different features in the classification process.

- Least Absolute Shrinkage and Selection Operator (LASSO): LASSO is particularly effective for feature selection in high-dimensional textual data, choosing only the most crucial features. Given that textual data often exhibits high multicollinearity, LASSO can address this by selecting a subset of correlated features while reducing others to zero.
- Debiased LASSO: The debiased LASSO employs a two-step approach to minimize bias by initially obtaining LASSO estimates and subsequently debiasing them. This method is especially valuable in textual data where precise feature importance is critical.

5.1.4 Model Evaluation

For evaluating the performance of these machine learning algorithms, we utilized 10-fold cross-validation. The training data was divided into ten parts, with each part serving once as a validation set while the remaining nine were used for training. Key performance metrics such as Accuracy, Precision, Recall, and F1 score were considered, each chosen to align with the specific needs of our study:

- Recall: This metric, representing the true positive rate, is critical for identifying as many scammers as possible to protect genuine users.
- Precision: High precision is essential to avoid mistakenly labeling genuine users as scammers, thereby ensuring fairness.
- F1 Score: The harmonic mean of precision and recall, the F1 score, helps balance these two metrics, ensuring neither is disproportionately favored, crucial for maintaining a positive user experience on the dating platform.

We employed Naive Bayes as a baseline model due to its quick training and effectiveness with categorical data. Two simple approaches were used for training:

1. Training and prediction using one-hot encoded categorical data only (attributes such as 'ethnicity', 'occupation', 'status', 'age group', 'country').
2. Training and prediction with a Bag-of-Words model on the profile descriptions.

5.2 Image

5.2.1 Data Preparation

- Image Preprocessing:
    - Normalization of Pixel Values: We scaled the pixel values of each image from the range [0, 255] to a normalized range of [0, 1]. This normalization facilitates numerical stability and expedites convergence during the training phase.
    - Resizing: To enhance computational efficiency, all images were resized to a uniform dimension of 128x128 pixels. This step is crucial for reducing the computational demand and streamlining the network architecture, especially when handling large datasets.
    - Grayscaling: To concentrate the model's learning on structural features rather than color, images were converted from RGB to grayscale by averaging the intensity values.

5.2.2 Feature Engineering

To enhance the model's ability to generalize across unseen images and to mitigate overfitting, we implemented several advanced feature engineering and fine-tuning techniques:

1. Data Augmentation: To ensure the model's robustness and its ability to generalize across new images, we employed data augmentation using the ImageDataGenerator class from Keras. This technique enriches the training dataset by applying various transformations such as rotations, scaling, translations, flipping, and shearing, thereby mimicking diverse real-world scenarios.
2. Gray-scaling Processing: Although initially processed into grayscale to simplify data, for the VGG-16 based model, we preserved the original RGB color channels. This adjustment allows the model to leverage the pre-trained network weights which are optimized for color information, enhancing feature recognition capabilities.
3. Batch Normalization and Dropout: Each convolutional layer was followed by batch normalization to stabilize neural network activations. In conjunction, dropout was implemented as a regularization technique, randomly deactivating a fraction of neurons during training to mitigate overfitting.
4. Callbacks for Training Efficiency: We integrated several Keras callbacks to refine the training process:
    - Early Stopping: This callback terminates training when there is no improvement in validation loss, preventing overfitting.
    - Reduce LR on Plateau: Automatically reduces the learning rate when learning plateaus, aiding in continued progress.
    - Model Checkpoint: Saves the model at predetermined intervals, allowing for recovery and resumption without loss of progress.

5.2.3 Model Selection

For our image classification task, we employed a Convolutional Neural Network (CNN) architecture, widely acclaimed for its ability to efficiently process structured grid data such as images. CNNs are particularly proficient at maintaining spatial relationships between pixels and learning hierarchical patterns through successive layers.

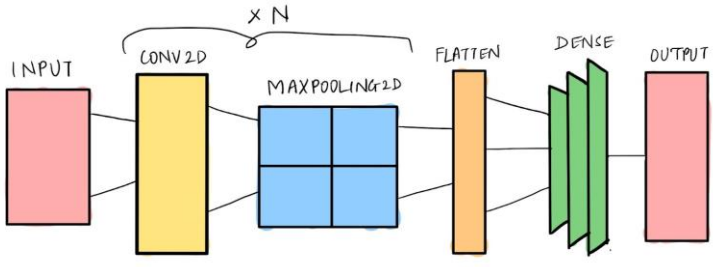

Fig 1: Convolutional Neural Network Basic Architecture

The CNN architecture was implemented in two approach directions:

1. **CNN with Multiple Convolutional Layers:** The architecture began with a single convolutional layer and expanded to four layers to capture increasingly complex features.

Each convolutional layer, employing a 3x3 filter with a stride of 1 and 'same' padding, used the ReLU activation function to introduce non-linearity. Following each convolutional layer, batch normalization facilitated faster convergence, while 2x2 MaxPooling layers reduced dimensionality. Dropout at a rate of 0.5 was used extensively to combat overfitting. The network concludes with two densely connected layers leading into a sigmoid output layer for binary classification(Fig 2).

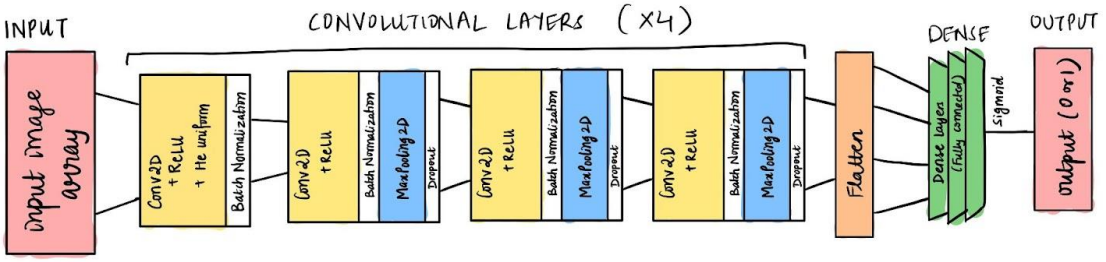

Fig 2: 4-Layer CNN Model Implementation

2. **VGG-16 Pre-Trained Hybrid Model:** Simultaneously, we adopted a hybrid approach using the VGG-16 architecture (Fig 3), pre-trained on the ImageNet dataset. This model uses transfer learning to harness a vast array of pre-learned features, significantly boosting its ability to discern between real and fake images. We kept the pre-trained layers frozen to preserve their learned characteristics, while additional custom densely connected layers were introduced to refine these features for our specific classification task.

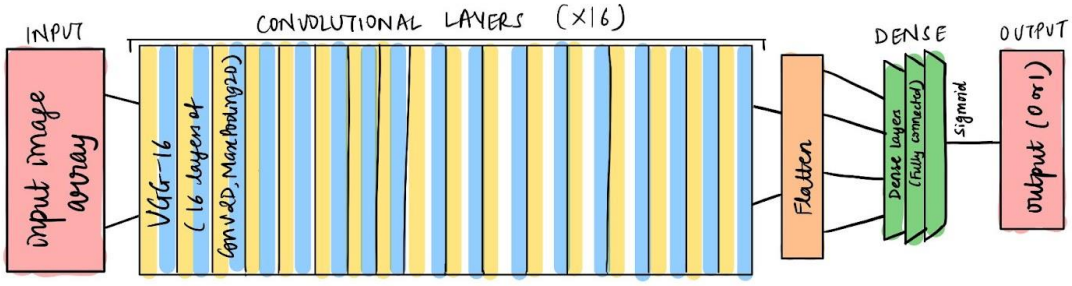

Fig 3: VGG-16 Hybrid Model Implementation

5.2.4 Model Evaluation

Both models were trained using the binary cross-entropy loss function and optimized with the Adam optimizer, with a learning rate set at 0.0001. The training was conducted over 30 epochs with a batch size of 32. Both training and validation losses were closely monitored throughout the training process to optimize parameters and mitigate the risk of overfitting. Model performance was assessed based on accuracy and binary entropy loss, which are crucial for accurate differentiation between real and AI-generated faces.

5.3 Combined Dataset:

5.3.1 Data Preparation

After thorough cleaning and preparation of the profiles and images, we integrated these into four distinct testing scenarios to evaluate the models' effectiveness:

1. Scam profile + AI-generated face
2. Scam profile + Real face
3. Real profile + AI-generated face
4. Real profile + Real face

Statistics from dating apps indicate that approximately 10% of users are scammers, a ratio we replicated in our test set. Scenarios 1, 2, and 3 were designated as "scams," given the presence of at least one deceptive element. We initially prepared a test subset comprising 598 real profiles and 596 scam profiles. We then combined these profiles with images in various scenarios: 19 real profiles with 19 AI-generated faces, 19 scam profiles with 19 real faces, and 19 scam profiles with 19 AI-generated faces. This resulted in a final test dataset of 636 rows, with 57 (~9%) labeled as scams.

6. Results

6.1 Profiles

6.1.1 Model Results

The Naive Bayes model delivered competitive results, prompting us to evaluate the remaining three classifiers similarly. Of these, the Logistic Regression (LR) model demonstrated superior accuracy and F1-score, particularly when employing the first approach, as detailed in Table 1. Consequently, we opted for the LR model for further experimentation. Detailed results of the subordinate models are provided in the Tables section (Appendix B).

Table 1: Baseline Model using Logistic Regression:

Feature set	Accuracy	Precision	Recall	F1	FNR	FPR
Categorical Features only	**0.900732**	0.878582	0.928680	0.902861	**0.035391**	0.063876
Bag Of Words (50 Words)	0.772974	0.832127	0.681986	0.749004	0.158333	0.068694
Bag Of Words (100 words)	0.793920	0.839716	0.724773	0.777225	0.137183	0.068897
Bag Of Words (500 words)	0.842301	0.869396	0.805493	0.835269	0.097183	0.060516
Bag Of Words (1000 words)	**0.852775**	0.874918	0.822649	0.847284	0.088595	**0.058630**
Bag Of Words (all)	0.852775	0.874918	0.822649	0.847284	0.088595	0.058630

Table 2: Logistic regression with TF-IDF

Feature set	Accuracy	Precision	Recall	F1	FNR	FPR
TD-IDF(50)	0.760626	0.763696	0.752187	0.757268	0.123567	0.115807
TD-IDF(100)	0.788894	0.794684	0.777208	0.785132	0.111215	0.099890
TD-IDF(500)	0.846280	0.853286	0.835205	0.843737	0.082099	0.071620
TD-IDF(1000)	**0.854246**	0.857509	0.849062	0.852633	0.075187	**0.070568**
TD-IDF(ALL)	0.839383	0.785505	0.932195	0.852154	**0.033719**	0.126898

Table 3: Fine Tuning with Tukey Method

	Accuracy	Precision	Recall	F1	AUC	FPR	FNR
count	10.000000	10.00000	10.000000	10.00000	10.00000	10.00000	10.00000
mean	**0.939474**	0.940723	0.937434	0.938921	0.983659	**0.029319**	**0.031208**
std	0.008661	0.011765	0.019055	0.009580	0.004076	0.005488	0.009660
min	0.918239	0.918455	0.898305	0.915767	0.974998	0.023013	0.016736
25%	0.938254	0.935223	0.928834	0.935736	0.982247	0.024618	0.024618
50%	0.941423	0.942729	0.940277	0.939895	0.984852	0.029350	0.029319
75%	0.943485	0.949551	0.949389	0.944981	0.986696	0.030922	0.036688
max	0.947699	0.954918	0.963964	0.949290	0.987168	0.039749	0.050314

Table 4: Fine Tuning LR with Sentence Embedding

	Accuracy	Precision	Recall	F1	FPR	FNR
count	10.000000	10.000000	10.000000	10.000000	10.000000	10.000000
mean	0.906178	0.893915	0.920321	0.906871	0.108005	0.079679
std	0.013136	0.015529	0.016283	0.014028	0.015013	0.016283
min	0.882600	0.871901	0.894068	0.882845	0.090517	0.057143
25%	0.896967	0.881356	0.907345	0.898775	0.098990	0.066560
50%	0.908996	0.891244	0.922343	0.905129	0.105084	0.077657
75%	0.917759	0.908569	0.933440	0.919922	0.112021	0.092655
max	0.920335	0.915323	0.942857	0.924000	0.137168	0.105932

Notably, as the number of words considered in the algorithm increased, all metrics generally improved. The Logistic Regression model, enhanced with the Tukey method, emerged as the most effective, achieving an accuracy of 93.9%, with a false negative rate of 3.12% and a false positive rate of 2.93%.

6.1.2 Analysis of Best Performance Model

For the textual data, our analysis was guided by two hypotheses:

1. Scammers attempt to mimic demographics perceived as most vulnerable.
2. Scammers employ similar descriptions to entice their targets.

Our initial analysis focused on the weights of the predictors within the LR model, trained solely on categorical data. The top 10 predictors with the highest absolute weights (Figure 1) provided key insights. For instance, the substantial weight of "status_widowed" indicates that profiles listing this status are significantly more likely to be scams, with the log odds ratio increasing ninefold, assuming all other factors remain constant. This finding supports our hypothesis, suggesting that scammers commonly list their status as 'widowed', their occupation as 'military', and fall within the age group of 21-30 years. In contrast, profiles less likely to be scams tend to belong to the age group of 71-80, identify as Hispanic, and originate from Colombia.

	features	coef		features	coef
59	status_widowed	3.311875	66	age_group_71-80	-3.112230
40	occupation_military	3.028107	8	ethnicity_hispanic	-3.051794
61	age_group_21-30	2.408689	90	country_Colombia	-2.860347
112	country_Ghana	2.316542	203	country_Venezuela	-2.492584
200	country_United states	2.221810	157	country_Peru	-2.342044
198	country_United Kingdom	1.725911	38	occupation_manufacturing	-2.314245
3	ethnicity_Native American	1.702394	100	country_Ecuador	-2.252582
199	country_United States	1.656198	43	occupation_repair	-2.117568
30	occupation_engineering	1.586845	44	occupation_retired	-2.101006
183	country_Switzerland	1.580306	144	country_Mexico	-2.061451

Fig 4: Top 10 for categorical features using LR model, positive weights (left) negative weights (right)

The distribution of predictor weights is illustrated in Figure 5, displaying a bell-shaped curve. With the interquartile range between -0.452 and 0.614 and the 95th percentile at 1.5, weights exceeding 1.5 are deemed significant. This aligns with our observations of scam profiles.

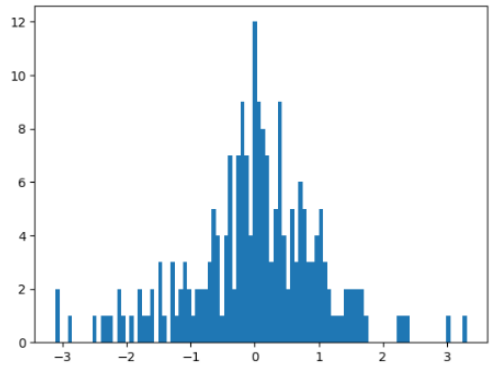

Fig 5: Plot of LR model weights

To explore the second hypothesis, we employed Sentence-BERT to transform descriptions into embedding vectors, which were then clustered using k-means. Despite experimenting with various cluster sizes and assessing with the silhouette score, the optimal number of clusters was found to be two. However, the clustering did not reveal a distinct pattern among scammer descriptions, as shown in Figure 6.

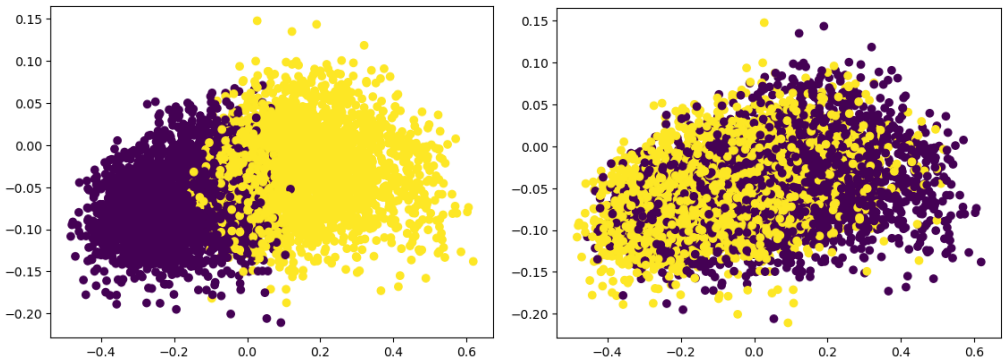

Fig 6: Cluster label(left), real labels(right)

Further analysis was conducted using a Bag of Words (BoW) approach on the descriptions, which resulted in a heavily right-skewed distribution of weights (Figure 7). Applying Tukey's method, we adjusted 'k' to 15 to accommodate the wide spread of data, identifying key phrases and words characteristic of scammer communications. Despite the lack of clustering in descriptions, the integration of these indicative words into the initial categorical data improved the results marginally, suggesting that while scammers may not use identical descriptions, certain phrases and words are commonly employed.

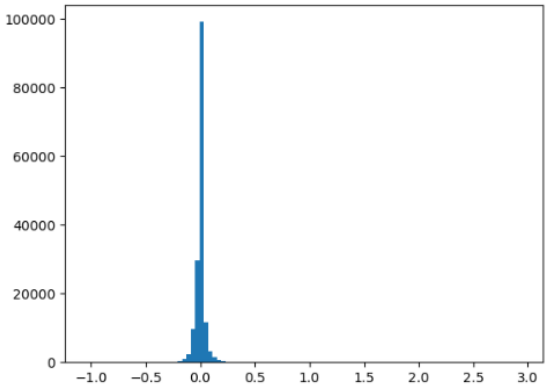

Fig 7: Plot of the BoW weights

	features	coef		features	coef
20617	com	2.937682	151218	wants	-1.042409
16621	caring	2.228260	88211	man looking	-0.911962
24409	cool	1.794866	95353	music	-0.904478
96149	name	1.711856	157198	worker	-0.898914
139048	tell	1.651552	3202	affectionate	-0.819325
127277	simple fun	1.497557	23016	consider	-0.796155
84475	lovely	1.477340	90201	mature	-0.783081
159011	yahoo	1.302606	84613	lover	-0.772609
87449	man	1.298761	57989	hello	-0.744603
137419	swimming	1.184834	77261	likes	-0.731670

Fig 8: Top 10 features for unigrams using LR model, positive weights (left) negative weights (right)

6.1.3 Further Analysis using LASSO/Debiased LASSO

In our initial trials with regular LASSO regression, the model underperformed, achieving an accuracy of only 67%. Recognizing the potential issues with overly aggressive feature selection leading to underfitting, we incorporated instrumental variables for double selection to debias the model. This adjustment significantly improved accuracy to 90%.

Table 5: lasso Regression Results

	LASSO Regression	Debiased LASSO
Accuracy	0.669179229480737	0.8994974874371859
Precision	0.6830601092896175	0.9018233658701996
Recall	0.6291946308724832	0.8994974874371859
F1 Score	0.6550218340611353	0.8997674889595019

This improvement suggests that the regular LASSO may discard crucial features along with noise, detrimentally simplifying the model. By utilizing double selection with instrumental variables, we lowered the regularization parameter, which enhanced the model's generalization capabilities on unseen data and corrected biases inherent in LASSO's initial feature selection approach.

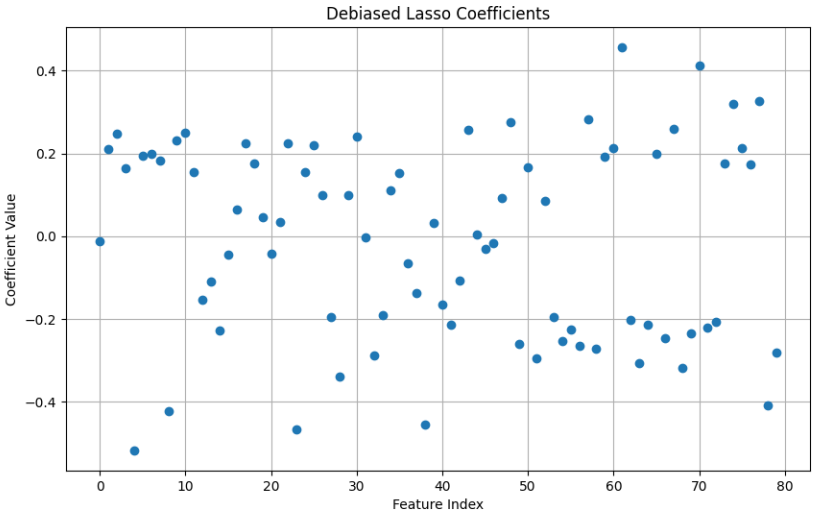

Fig 9: Scatter Plot for Lasso Coefficients

The scatterplot (Fig 9) illustrates the variance in coefficient values, indicating no single feature's overwhelming influence in the model. Ultimately, the debiased LASSO model demonstrated robust predictive performance, marked by high accuracy, precision, recall, and F1 score.

6.2 Image

6.2.1 Model Results

The evaluation of our image classification model began with assessing a subset of the data using accuracy as the primary metric (Table 6). This preliminary step helped identify the most promising models, which were then tested across the full dataset (Table 7). Along with accuracy, we utilized a confusion matrix and ROC curve to thoroughly evaluate the training and validation performance of the selected model. Detailed cross-entropy graphs for the other models are available in the Figures section (Appendix D).

Table 6: Test Accuracy% (training: 2400, validation: 800, testing: 800)

Input		Model Type	
		4-CNN	**VGG-16**
Without Grayscale Images (RGB)	With Data Augmentation	74.00%	73.12%
	Without Data Augmentation	71.63%	77.37%
With Grayscale Images (Grayscale)	With Data Augmentation	77.38%	
	Without Data Augmentation	99.50%	

Table 7: Test Accuracy% (training:16,000, validation: 2000, testing: 2000)

Input		Model Type	
		4-CNN	**VGG-16**
Without Grayscale Images (RGB)	With Data Augmentation	85.7%	84.1%
	Without Data Augmentation	85.2%	83.7%
With Grayscale Images (Grayscale)	With Data Augmentation	88.0%	
	Without Data Augmentation	**100%**	

Our assessment of different models highlighted the 4-CNN model as the most accurate, particularly when using grayscale data. This suggests that grayscaling enhances performance for basic CNN architectures. Conversely, the VGG-16 Hybrid model underperformed for our specific task, likely due to its training on a broad feature set not optimally suited for this narrower application. Additionally, signs of overfitting were observed towards the end of the training period, which hindered its performance on unseen test data.

Evaluation Rationale for chosen (best) model

During the initial training phases, the 4-CNN model showed promising signs of learning, with rapid improvements in training accuracy and loss. However, fluctuating validation metrics, like the drop in accuracy to 76.02% at epoch 3 despite high training accuracy of 99.16%, indicated potential overfitting. Adjustments to the learning rate helped stabilize the model, enhancing its ability to generalize as seen in the convergence of training and validation metrics.

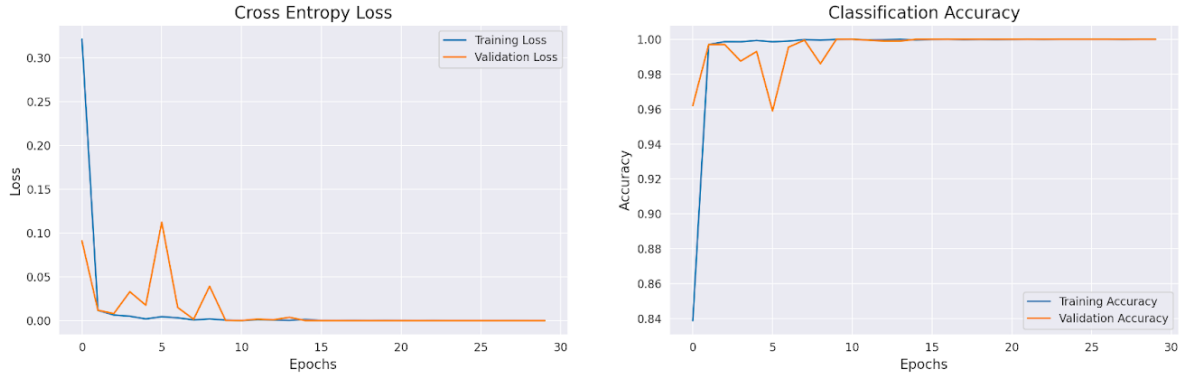

Fig 10: Cross Entropy for best model: 4-CNN with grayscale data without data augmentation

6.2.2 Analysis of Best Performance Model

We hypothesized that AI-generated images and digitally altered photographs could be identified by discerning uniform patterns typical to image manipulations. Key traits observed in manipulated

images included overly polished appearances, unnatural backgrounds, absence of natural shadows, and discrepancies in image quality.

To validate our model's decisions, we used the Local Interpretable Model-agnostic Explanations (LIME) technique, which helped visualize influential image areas:

- Green Areas: Indicate positive contributions to the prediction, such as smooth skin textures or stock-like backgrounds.
- Red Areas: Represent features that detract from the prediction, typically aligning with natural imperfections and interactive backgrounds in genuine images.
- Gray Areas: Neutral, indicating less informative or ambiguous features.

The effectiveness of the model was further analyzed using LIME on two types of images, categorized by different levels of prediction confidence:

1. High Confidence Cases:

Images where the model showed strong conviction in its classification (which was correct) evidenced by prediction probabilities near 0 or 1.

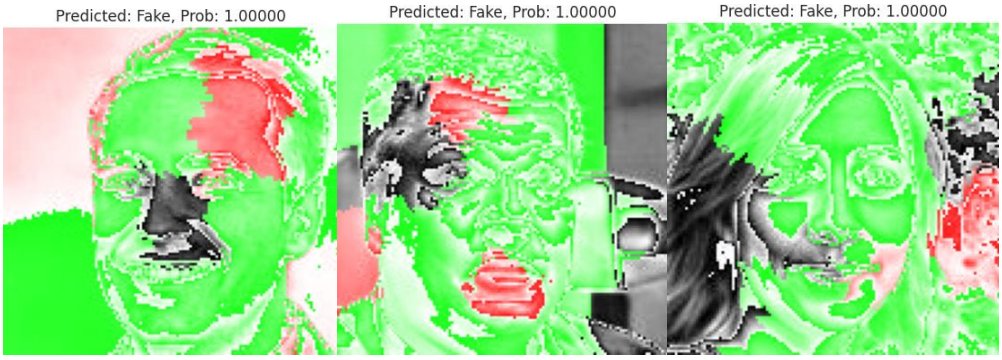

Fig 11: LIME Images for High Confidence: Predictions probabilities closest to 0 or 1

Two particular features significantly influence the model's decision-making:

- Background Analysis: The model notably focuses on the backgrounds, where green regions often indicate digitally manipulated or inconsistent backdrops suggesting superimposition. This pattern underscores the model's ability to detect anomalies in background uniformity and depth as indicators of a fake image.
- Facial Analysis: Green areas around facial folds and contours, especially where shadows appear manipulated, play a crucial role in classification. The model uses these altered shadows as cues for identifying digital modifications, categorizing such images as 'fake'.

2. Edge Cases:

With predictions close to the decision threshold of 0.5, these images were ambiguous and challenging for the model to classify with high confidence.

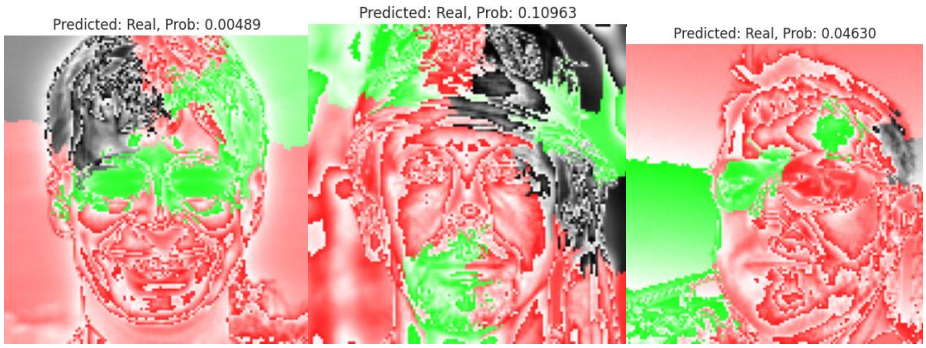

Fig 13: LIME Images for Edge cases

Images near the decision threshold (0.5) present a challenge, displaying a mix of green and red areas across faces and backgrounds. This mixture indicates a conflict in feature interpretation, with competing features influencing the model almost equally. Such cases often involve subtle manipulations or lower quality images, complicating definitive predictions. In these instances, gray areas become more prevalent, signifying parts of the image that do not significantly impact the model's decision.

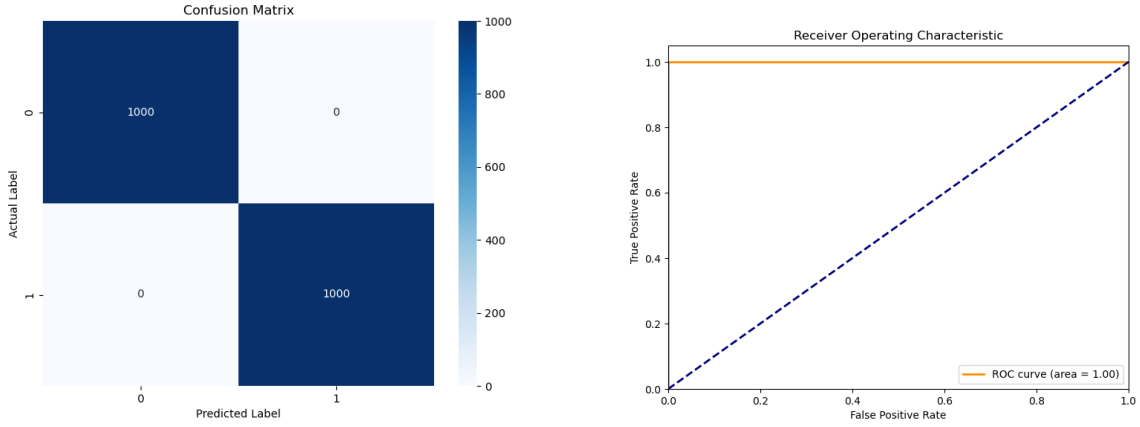

Fig 14: Confusion matrix (left) and ROC curve (right) for best model on test data

The confusion matrix (Fig 13 left) highlights the model's proficiency, showing that it correctly identified all 2000 test images as genuine (true positives). This high success rate in detecting genuine images demonstrates the effectiveness of our feature selection and the robustness of the model.

Moreover, the ROC curve, with an area under the curve (AUC) of 1.00, confirms the model's exceptional ability to perfectly distinguish between genuine and manipulated images, achieving 100% sensitivity and specificity. These results illustrate the model's strong discriminatory power and validate the reliability of our analytical approach.

6.3 Combined Dataset

After developing the profile and image models separately, we explored two methods to combine their predictions:

1. Setting up an OR gate where the profile is classified as scam if at least one model predicts scam
2. Combining the models via a weighted average

The benefit of the OR gate is that the 2 models would not interfere with each other's predictions of scam or not. The decision boundary would also be the same as our definition of scam which is at least a single factor of untruth.

By setting the threshold of at 0.8 for each model, we were able to obtain a result of:

Accuracy	Precision	Recall	False positive rate	False negative rate
0.963836	0.750000	0.894737	0.029361	0.105263

Table 8: Threshold for each model set at 0.8

Combining the predicted probabilities via a weighted average provides the possibility of giving one model a higher importance than the other. This weight can be tuned according to the subject matter experts' recommendations. For this project, we decided to test out 2 different variations of the weighted average:

1. Naive method: Setting the weights to be 0.5 for each of the models
2. Assigning a higher weightage to the Image model (0.6) as results indicated higher accuracy as compared to the profile model (0.4).

By setting a low threshold for the combined probabilities at 0.3, we ensure that both profile and image model predicts a low probability of scam before classifying the user as non-scam.

Accuracy	Precision	Recall	False positive rate	False negative rate
0.915094	0.514563	0.929825	0.086356	0.070175

Table 9: Setting weights to be equal

Accuracy	Precision	Recall	False positive rate	False negative rate
0.949686	0.662338	0.894737	0.044905	0.105263

Table 10: Higher weightage to image model

6.4 Future Considerations

Given above results, we would like to suggest some future improvements:

1. Dataset Expansion:

a. Include additional scam sources to enrich the dataset and capture specific features missed in the current data.
b. Enhance model generalizability by incorporating diverse scam profiles and images.

2. Enhanced Model Training:
   a. Utilize more computational resources and time to train pre-trained models on larger datasets with varied data and training cycles.
   b. Aim for better model generalization capabilities to improve real-world applicability.
3. Exploring Ensemble Methods:
   a. Investigate other ensemble methods such as stacking or hierarchical fusion to enhance model performance.
   b. Understand the relationship between images and textual data for more insightful feature fusion.

7. Conclusion

7.1 Summary of Results

The combined model achieved an impressive accuracy of 96.3%, with a false positive rate of 2.9% and a false negative rate of 10.3%. While the individual models showed strong performance, with the textual model achieving 93.9% accuracy and the image model achieving 100% accuracy, there is still considerable room for improvement, especially in optimizing the integration of both models.

7.2 Hypothetical Data Pipeline

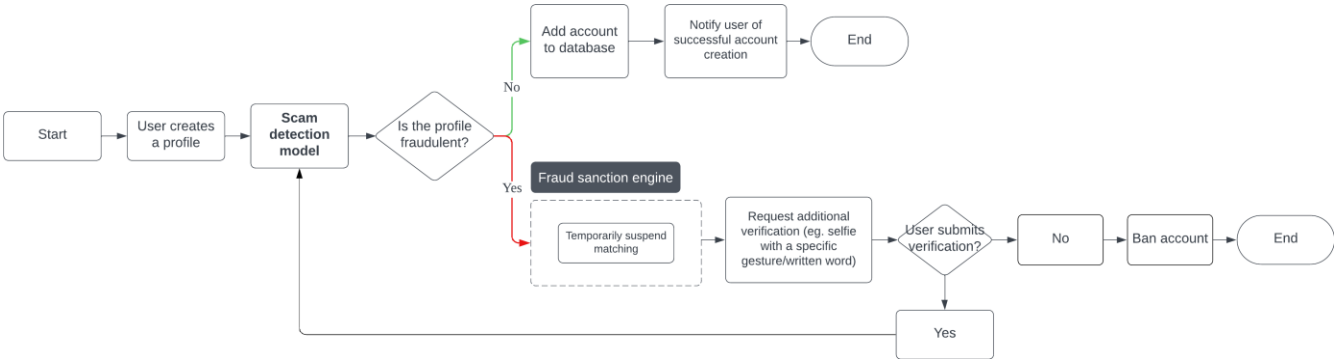

Fig 15. Hypothetical data pipeline for dating app company

7.3 Recommendations

The developed fraud detection model, tailored for dating websites, offers valuable insights and solutions to enhance user authenticity and combat fraudulent activities. Beyond dating platforms, it can be seamlessly integrated into various social media channels to cross-reference profile information and validate user identities based on their online presence. This proactive approach

can significantly mitigate risks associated with fake accounts and identity theft across digital platforms.

Moreover, extending the application of this model to e-commerce platforms and online marketplaces like Carousell presents a promising opportunity. By verifying seller profiles and scrutinizing product listings, the model can effectively curb counterfeit dealings and enhance consumer trust in online transactions.

8. Appendix

Appendix A: References

- FTC's Office of Technology. (2024). As nationwide fraud losses top $10 billion in 2023, FTC steps up efforts to protect the public. Retrieved from https://www.ftc.gov/news-events/news/press-releases/2024/02/nationwide-fraud-losses-top-10-billion-2023-ftc-steps-efforts-protect-public
- Chua, N. (2023, February 9). Scam victims in s'pore lost $660.7m in 2022; more than half of them were young adults. The Straits Times. https://www.straitstimes.com/singapore/scam-victims-in-s-pore-lost-6607-million-in-2022-almost-13-billion-in-past-two-years
- Tan, S. (2024, February 7). Match, Chat, Love: Examining the popularity and usage of dating apps in Singapore. YouGov. https://business.yougov.com/content/48571-match-chat-love-examining-the-popularity-and-usage-of-dating-apps-in-singapore
- Complete Real Images Dataset (Flickr FacesHQ): https://drive.google.com/drive/folders/1u2xu7bSrWxrbUxk-dT-UvEJq8IjdmNTP
- Complete Fake Images Dataset (1 million fake faces): https://archive.org/details/1mFakeFaces
- Strandell, J. (2024) *Welcome to the age of fake dating profiles – welcome to the age of fake dating profiles*, *Besedo*. Available at: https://besedo.com/knowledge-hub/blog/welcome-to-the-age-of-fake-dating-profiles/#:~:text=The%20problem%20with%20fake%20profiles%20is%20more%20widespread%20than%20you,not%20even%20a%20real%20person.

Appendix B: Data Dictionary

Dataset Name: final_test_dataset.csv
Path: DSA4263_Dating_Fraud/data/processed/
Number of Attributes: 11
Input Data: pre-processed profiles + images

Table 8: Data Dictionary for combined test data

Sr. No.	Attribute Name	Data Type	Description

1	image_path	string	unique file path for each image
2	face_fake	binary	label for each image: Real = 0 ; Fake = 1
3	age	integer	Age of profile holder
4	location	string	City/Region, Country of Residence of profile holder
5	ethinicity	string	Race of the profile holder
6	occupation	string	Job Profession of profile holder
7	status	string	Marital status of profile holder
8	description	string	personal introduction/description of profile holder
9	scam	binary	Label for each profile: Real = 0 ; Fake = 1
10	age_group	string	Age group of profile holder in bins
11	country	string	Extracted country of residence of profile holder

Appendix C: Tables

Profiles

Table 9: Baseline Model using naive bayes

Feature set	**Accuracy**	**Precision**	**Recall**	**F1**	**FNR**	**FPR**
Categorical Features only	0.889843	0.85383	0.939604	0.894507	0.029947	0.080209
Bag Of Words (50 Words)	0.69361	0.692239	0.699875	0.694296	0.149749	0.156634
Bag Of Words (100 words)	0.732561	0.729197	0.741683	0.733578	0.129233	0.138207
Bag Of Words (500 words)	0.798752	0.781027	0.830387	0.803946	0.084616	0.116632
Bag Of Words (1000 words)	0.816132	0.798264	0.846432	0.820689	0.076657	0.107212
Bag Of Words (all)	0.82807	0.771079	0.931284	0.843271	0.034136	0.137788

Table 10: Baseline Model using SVM

Feature set	**Accuracy**	**Precision**	**Recall**	**F1**	**FNR**	**FPR**
Categorical Features only	0.893606	0.864085	0.933595	0.897289	0.0330923	0.073302

Bag Of Words (50 Words)	0.772348	0.843594	0.666896	0.744258	0.165872	0.061781
Bag Of Words (100 words)	0.792029	0.847775	0.710744	0.772437	0.144307	0.063665
Bag Of Words (500 words)	0.836854	0.861719	0.802229	0.829818	0.098859	0.064286
Bag Of Words (1000 words)	0.846704	0.868511	0.816326	0.840895	0.091736	0.061560
Bag Of Words (all)	0.843351	0.855138	0.826689	0.839810	0.086498	0.070150

Table 11: Baseline Model using Random Forest

Feature set	**Accuracy**	**Precision**	**Recall**	**F1**	**FNR**	**FPR**
Categorical Features only	0.892355	0.882155	0.904627	0.893002	0.047327	0.060318
Bag Of Words (50 Words)	0.785757	0.800156	0.758398	0.778481	0.120214	0.094029
Bag Of Words (100 words)	0.804605	0.799652	0.810303	0.804502	0.094241	0.101154
Bag Of Words (500 words)	0.854246	0.847049	0.863010	0.854504	0.068063	0.077692
Bag Of Words (1000 words)	0.863460	0.854615	0.874965	0.864120	0.062201	0.07433
Bag Of Words (all)	0.865762	0.859347	0.873711	0.865926	0.062829	0.071409

Appendix D: Figures

Image Analysis (Sub-optimal) Model Runs on complete data

Fig 16: Cross Entropy for VGG-16 Hybrid Model with Data Augmentation

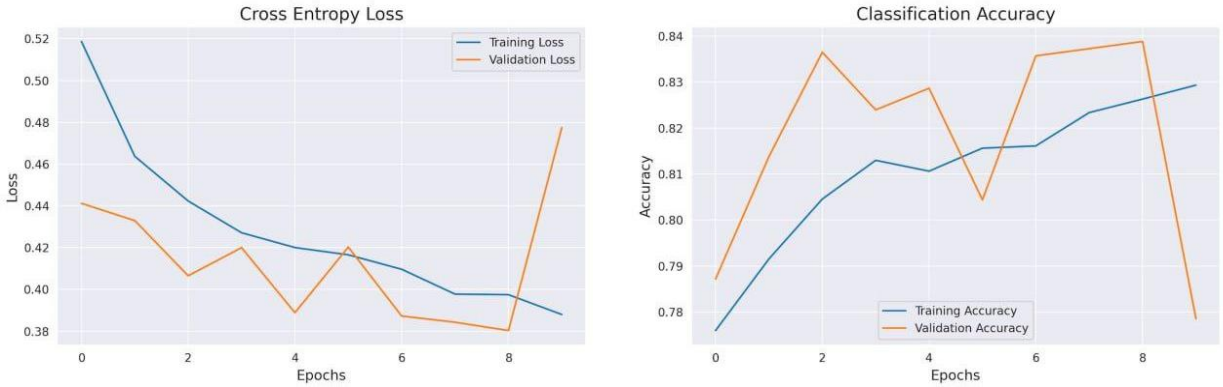

Fig 17: Cross Entropy for VGG-16 Hybrid Model without Data Augmentation

Fig 18: Cross Entropy for 4-CNN Base Model with Data Augmentation (without grayscaling)