ENPM703 Fundamentals of AI and Deep Learning
Fall 2025

Project Report
Group 3

Phishing Email Detection with Multimodal Deep Learning

Vishal Patil (121372253)
Srihari Narayan (121386151)
Akash Vora (122094499)
Anila Sai Namburi (121387158)

UMD Honor Pledge

"I pledge on my honor that I have not given or received any unauthorized assistance on this assessment."

Abstract

Phishing attacks exploit deceptive email content and fraudulent brand logos to compromise user security. This project develops a comprehensive multimodal machine learning system that detects phishing emails by jointly analyzing textual content, and visual logo elements. We implement and compare six distinct models: K-Nearest Neighbors, Logistic Regression, custom convolutional neural networks for text and image analysis, pretrained ResNet18 for logo classification, and a novel dual-tower fusion architecture integrating all modalities. Using a combined email dataset of 76,346 samples and the OpenLogo dataset containing 72,652 brand logos across 352 classes, our models achieve validation accuracies ranging from 76.30% to 98.96%, and our dual-tower fusion model, which successfully integrates text features (256-dim), image features (512-dim), to provide robust multimodal phishing detection that surpasses single-modality approaches, has the highest validation accuracy at 99.45%.

1. Introduction

Phishing emails remain one of the most prevalent cybersecurity threats, responsible for billions of dollars in financial losses annually. These attacks employ sophisticated social engineering techniques, combining deceptive textual content with fraudulent brand logos to manipulate recipients into divulging sensitive information or downloading malware. Traditional rule-based detection systems struggle to keep pace with evolving attack strategies, necessitating more adaptive machine learning solutions.

1.1 Problem Statement

Current phishing detection systems predominantly rely on either textual analysis or simplistic heuristics, failing to capture the full complexity of modern phishing attempts. Attackers increasingly leverage legitimate-looking brand logos and carefully crafted language to bypass filters. Our project addresses this gap by developing a multimodal detection system that simultaneously analyzes email text and embedded logos to identify phishing attempts with high accuracy.

1.2 Key Contributions

This project makes three primary contributions:

1) Comprehensive Baseline Comparison: We implement and evaluate multiple model architectures (KNN, custom CNNs, pretrained ResNet18) across different modalities to establish performance baselines.
2) Multimodal Fusion Architecture: We design a novel dual-tower fusion model that integrates text and image features through learned representations.
3) Practical Deployment System: We develop an end-to-end inference pipeline capable of processing raw email HTML and extracting relevant features for real-time classification.

1.3 Performance Overview

Our models achieve the following validation accuracies:

- K-Nearest Neighbors (text): 81.71%
- Logistic Regression (text): 80.00%
- Custom Text CNN: 98.96%
- Custom Image CNN: 76.30%
- ResNet18 Image (pretrained): 97.43%
- Dual-Tower Fusion (Custom Text CNN + Custom Image CNN): 99.45%

2. Related Work

Phishing Detection Methods: Traditional phishing detection relies on blacklists, URL analysis, and simple keyword matching. Recent work by Doshi et al. [1] proposed a dual-layer architecture combining content-based and URL-based features, achieving 95% accuracy. However, their approach treats text and images independently without true multimodal fusion.

Text-Based Approaches: Zhang and Wallace [4] demonstrated the effectiveness of convolutional neural networks for sentence classification tasks. Their work showed that simple CNN architectures with multiple filter sizes can capture meaningful textual patterns. We adapt

their approach for phishing-specific language detection.

Image-Based Logo Detection: The OpenLogo dataset [2] provides a comprehensive benchmark for logo recognition with 352 brand classes. Simonyan and Zisserman [3] introduced VGG-style architectures demonstrating that depth is critical for visual recognition tasks. We leverage both custom VGG-style networks and pretrained ResNet18 for logo classification.

Differences from State-of-the-Art: Unlike prior work focusing on single modalities, our system performs joint training on text and image features. Our dual-tower architecture with learned fusion weights enables the model to dynamically weight different modalities based on their discriminative power for each sample. Additionally, we incorporate metadata features including sender information, URL counts, and text statistics that prior work has overlooked.

3. Data

3.1 Text Dataset

We compiled a comprehensive email dataset combining multiple public sources:

- CEAS_08: Conference on Email and Anti-Spam 2008 challenge dataset
- Enron: Legitimate corporate emails
- Nazario: Curated phishing samples
- Nigerian Fraud: Advance-fee fraud emails

After preprocessing and deduplication, our final dataset contains 76,346 emails with binary labels (phishing/legitimate). Each email includes subject line, body text, sender, receiver, date, and extracted URLs, and additional metadata features. We standardized all sources into a unified schema ensuring consistency.

Data Split: 80% training (61,077 samples), 20% validation (15,269 samples). We ensured balanced class distribution across splits with approximately 48% phishing and 52% legitimate emails.

Key Statistics:

- Average email length: 287 words (legitimate), 156 words (phishing)
- Vocabulary size: approx. 200,000 unique tokens (after filtering words appearing <2 times)
- Maximum sequence length: 512 tokens (99th percentile coverage).

Beyond the text-only dataset, we also worked with emails in their native HTML format to support realistic deployment scenarios. These HTML files preserve the original structure including embedded images, styling, and metadata that plain text representations lose. To handle the scarcity of real phishing HTML samples, we augmented our dataset using generative AI to create realistic phishing HTML emails that maintain authentic structure and formatting patterns. This HTML dataset enables our inference pipeline to process emails as they naturally appear in user inboxes, extracting text content, and detecting embedded logos from the raw HTML structure.

3.2 Image Dataset

We utilize the OpenLogo dataset [2] containing 72,652 logo images across 352 brand classes. This dataset provides diverse logo variations including different resolutions, backgrounds, and aspect ratios, making it ideal for training robust logo classifiers.

Data Split: 80% training (58,122 images), 20% validation (14,530 images).

Preprocessing:

- All images resized to 224×224 pixels
- Normalized using ImageNet statistics (mean=[0.485, 0.456, 0.406], std=[0.229, 0.224, 0.225])
- Data augmentation: random horizontal flips, color jittering (brightness=0.2, contrast=0.2)

Class Distribution: The dataset exhibits class imbalance with popular brands having 500+ samples while others have around 50. We address this through weighted sampling during training.

3.3 Metadata Features

For the fusion model, we engineered a 20-dimensional metadata feature vector capturing:

- Email length statistics (source, sender, receiver, date, subject, body, urls, label, sender_domain, url_list, num_urls, url_domains, has_urgent_words, num_urgent_keywords, subject_length, body_length, sender_domain_suspicious, domain_brand_mismatch, receiver_domain, sender_receiver_same_domain, metadata_vector, mentioned_brands, num_brands_mentioned, has_brand_mention, image_brand_ids, has_image_mappable_brand)
- URL characteristics (count, presence of IP addresses, suspicious TLDs)
- Sender/receiver domain features
- Special character frequency
- Presence of urgency keywords ("verify", "urgent", "expire")
- HTML tag complexity metrics

These features provide complementary signals that pure text/image analysis might miss.

4. Methods

4.1 Custom Text CNN

Our custom Text CNN implements a one-dimensional convolutional architecture. The model begins with an embedding layer that maps the vocabulary to 128-dimensional dense vectors, learning semantic representations for each word. We then apply four convolutional blocks with progressively increasing channel dimensions of 64, 128, 256, and 512, allowing the network to learn increasingly abstract features. Each block consists of a Conv1D layer with kernel size 3, batch normalization for training stability, ReLU activation for non-linearity, and max pooling to reduce temporal dimensions while retaining salient features. Global average pooling reduces the final temporal dimension to a fixed-size representation regardless of input length. Fully connected layers then project from 512 dimensions through 256 dimensions to the final 2-class output for binary classification. This architecture learns hierarchical text representations where early layers capture character and word-level patterns while deeper layers capture semantic meaning and long-range dependencies.

4.2 Custom Image CNN

Our custom Image CNN implements a VGG-style architecture with four convolutional blocks designed to learn hierarchical visual features. The first block transforms the input 3-channel RGB image to 64 channels using two convolutional layers. The second block expands from 64 to 128 channels with two additional convolutional layers. The third block increases capacity further from 128 to 256 channels using three convolutional layers, capturing mid-level visual features. The fourth and final block expands from 256 to 512 channels with three convolutional layers, learning high-level semantic representations of logo characteristics. Each convolutional block is followed by 2×2 max pooling to progressively reduce spatial dimensions while increasing receptive fields. After the convolutional tower, global average pooling condenses spatial information into a single feature vector. Fully connected layers then project through dimensions 512, 512, 256 before reaching the final 352-class output for logo classification. This deep architecture with batch normalization and ReLU activations throughout enables learning discriminative logo features across hundreds of brand classes.

For comparative analysis, we implemented a ResNet18 baseline with ImageNet-pretrained weights to leverage transfer learning. We load the pretrained network and replace only the final fully connected layer to predict 352 classes. Since the pretrained backbone already captures rich visual features like edges, textures, and object parts learned from millions of images, we fine-tune all layers using a reduced learning rate (1e-4) to adapt to logo recognition while avoiding catastrophic forgetting. This setup significantly cuts training time and delivers higher accuracy than training from scratch.

4.3 Dual-Tower Fusion Architecture

Our novel fusion model integrates both modalities through a carefully designed dual-tower architecture. The text tower loads pretrained Custom Text CNN weights and extracts 256-dimensional features from the penultimate layer. This tower includes the embedding layer followed by four convolutional blocks, global pooling, and a fully connected projection to produce the 256-dimensional text feature vector.

The image tower loads pretrained Custom Image CNN weights and extracts 512-dimensional features using a similar approach. This tower implements four VGG-style convolutional blocks followed by global pooling and a fully connected projection producing the 512-dimensional image feature vector.

For metadata integration, we design a small multi-layer perceptron that projects the 20-dimensional metadata vector to 64 dimensions. This projection consists of a linear transformation from 20 to 64 dimensions, followed by ReLU activation and dropout with probability 0.25 to prevent overfitting on the metadata features. This transformation allows the network to learn non-linear combinations of metadata features that are most relevant for classification.

The fusion classifier concatenates all feature vectors into an 832-dimensional combined representation (256 text + 512 image + 64 metadata). This passes through four fully connected layers (832→512→256→128→2) with batch normalization, ReLU activation, and dropout (p=0.5) for regularization, producing the final binary classification output.

4.4 HTML-to-Decision Inference Pipeline

To support practical deployment in real-world email systems, we created a Python script that processes raw email HTML files to generate phishing predictions with confidence scores. The pipeline operates through four key stages:

HTML Parsing: The system extracts visible text while removing markup, analyzes hyperlink patterns, and computes HTML statistics such as tag complexity, JavaScript presence, and external resource counts.

Image Extraction: Multiple detection methods capture potential brand logos, including standard embedded images, and externally linked images from remote URLs.

Preprocessing: Text is tokenized using the training vocabulary, then truncated or padded to 512 tokens and converted to numerical IDs. Images are resized to 224×224 pixels and normalized with ImageNet statistics. Metadata

features from URLs and text statistics are scaled using training normalization parameters.

Model Inference: The trained dual-tower fusion model generates predictions using available modalities. When images are absent, the system uses a zero-filled dummy tensor for the image tower. Output includes the predicted class, a confidence score between 0 and 100 percent derived from softmax probabilities, and optional debugging signals like URL counts and suspicious TLD detection.

This pipeline integrates directly into email security infrastructure, processing native HTML formats without manual intervention. Additionally, we even deployed this aspect of the project with a GUI as a web application. We will go over this in the following section.

4.5 Code Architecture and Implementation Overview

The system implements a modular dual-tower architecture with three primary neural network components defined in train_fusion3.ipynb. The text tower, implemented as TextFeatureExtractor, accepts tokenized email text as integer sequences with shape [batch_size, 512] derived from the combined email dataset of 76,346 samples (CEAS_08, Enron, Ling, Nazario, Nigerian Fraud). The architecture consists of an embedding layer that maps vocabulary tokens to 128-dimensional dense vectors, followed by four 1D convolutional blocks with progressive channel expansion (64→128→256→512 channels), global average pooling, and fully connected projection. This tower outputs a 256-dimensional text feature vector capturing semantic phishing patterns and urgency language.

The image tower, implemented as ImageFeatureExtractor, processes logo images as tensors with shape [batch_size, 3, 224, 224] from the OpenLogo dataset containing 72,652 images across 352 brand classes. The architecture follows a VGG-style design with four 2D convolutional blocks (64→128→256→512 channels), each containing multiple Conv2D layers with batch normalization and ReLU activations followed by max pooling. Adaptive pooling and fully connected projection produce a 512-dimensional image feature vector encoding brand visual identity and detecting potential logo mismatches.

The fusion architecture, implemented in DualTowerFusionModel, concatenates all three feature vectors (256 + 512 + 64 = 832 dimensions) and processes them through a deep classifier with four fully-connected layers (832→512→256→128→2). Each layer includes batch normalization, ReLU activation, and dropout for regularization. The final layer outputs binary logits for phishing classification. The architecture supports two-stage optimization where fusion weights learn first from pretrained features, followed by end-to-end fine-tuning. Training relies on the golden dataset unified_multimodal_text.csv, which serves as the multimodal bridge. This dataset was constructed by standardizing raw email collections and engineering metadata features, while crucially scanning email text for known brand names to explicitly link them with specific logo image IDs. This mapping ensures the model trains on semantically aligned text-image pairings, allowing it to detect discrepancies between the email's stated intent and its visual branding.

4.6 Loss Function and Metrics

We use cross-entropy loss for all classification tasks, which provides strong gradient signals for probabilistic predictions and naturally handles the binary classification problem. For comprehensive evaluation, we report multiple metrics beyond simple accuracy. Accuracy measures overall classification correctness across both classes. Precision and recall specifically for the phishing class are critical metrics since minimizing false negatives (missing actual phishing emails) is paramount for security applications. The F1-score provides the harmonic mean of precision and recall, balancing both metrics into a single value. For the fusion model, we additionally compute ROC-AUC to analyze performance across different decision thresholds and understand the trade-off between true positive rate and false positive rate. We also report prediction confidence (the model's calibrated probability for the predicted class), which is especially useful on real-world HTML emails because phishing detection is inherently risk-based and often requires thresholding rather than a fixed yes/no decision. Confidence gives a graded estimate of certainty, enabling security teams to (1) tune decision thresholds to favor high recall (catch more phishing) or high precision (reduce false alarms), (2) prioritize triage by surfacing high-confidence phishing emails first while routing low-confidence cases for additional analysis or human review, and (3) monitor distribution shifts in deployment—systematic drops in confidence can signal that incoming email styles/content have changed and the model may need recalibration or retraining.

5. Experiments & Results

5.1 Training Configuration

All models were trained on NVIDIA GPU (GeForce RTX 4060 Laptop GPU) hardware using consistent hyperparameters within each model family to enable fair comparison. For text models, we used a batch size of 64 samples to balance GPU memory utilization and gradient estimation quality. Image models used a batch size of 32 due to higher memory requirements from spatial dimensions. The fusion model used a batch size of 16 to accommodate the memory overhead of processing three modalities simultaneously. We employed the Adam

optimizer with beta coefficients of 0.9 and 0.999 for all models, providing adaptive learning rates with momentum. Learning rates were set to 1e-3 for custom models training from scratch and 1e-4 for pretrained models or fusion training to prevent disrupting learned features. Weight decay of 1e-4 provided L2 regularization preventing overfitting. Training proceeded for 10 epochs for the independent text and image models, and 10 epochs for fusion tower as well.

5.2 Custom Text CNN Results

The custom Text CNN showed remarkable learning progression across 10 training epochs. At epoch 1, training accuracy reached 99.39% while validation accuracy was 98.04%, with training loss at 0.0203 and validation loss at 0.1220. By epoch 2, training accuracy increased to 99.68% and validation accuracy to 98.89%, with losses decreasing to 0.0114 and 0.0511 respectively. Training continued to improve steadily, reaching 99.86% training accuracy and 98.92% validation accuracy by epoch 7. The best performance occurred at epoch 10 where training accuracy reached 99.93% and validation accuracy peaked at 98.96%, our best result, with training loss reduced to just 0.0035 though validation loss increased slightly to 0.1059.

Validation accuracy plateaus near 98.96%, indicating minimal overfitting despite the model's high capacity of 3.2 million parameters. The small gap between training (99.93%) and validation (98.96%) accuracy suggests effective regularization through dropout and batch normalization. The model generalizes well as it maintains high validation accuracy throughout training.

Inspection of learned representations reveals that the model focuses on several key patterns' characteristic of phishing emails. Early convolutional layers activate strongly on urgency phrases such as "act now", "verify account", "limited time", and "expire". Middle layers detect the presence and patterns of URLs. Deeper layers capture grammatical inconsistencies common in phishing attempts including unusual capitalization, excessive punctuation, and non-native English language patterns.

5.3 Custom Image CNN Results

The custom Image CNN trained from scratch on logo classification showed steady improvement over 10 epochs. Starting from random initialization at epoch 1, training accuracy was only 12.24% while validation accuracy reached 20.81%, with high losses of 4.4316 and 3.8099. By epoch 3, accuracy had improved substantially to 39.58% training and 47.27% validation. Continued training brought epoch 5 results to 54.87% training accuracy and 61.21% validation accuracy with losses decreasing to 1.9397 and 1.6334. The model continued learning through epoch 10, achieving final results of 73.56% training accuracy and 76.30% validation accuracy with losses of 1.0859 and 0.9793 respectively. The validation accuracy of 76.30% at epoch 10 represents our best performance for this architecture.

The steady monotonic improvement across all epochs demonstrates that the custom CNN is effectively learning logo features despite training from scratch without pretrained weights. The consistent progression from random initialization (12.24%) to final performance (73.56% training, 76.30% validation) shows the model is successfully learning hierarchical visual representations. Error analysis reveals that most classification mistakes occur between visually similar brands. For example, the model confuses Pepsi and Coca-Cola due to their shared circular logo format with similar color schemes.

Comparative Analysis: ResNet18, which is a pretrained model, performed much better than the Custom Image CNN. The model reached its best performance at epoch 5 with 98.81% training accuracy and 97.43% validation accuracy, with training loss just 0.0610 and validation loss 0.1155. This substantial improvement validates the power of transfer learning for specialized visual tasks. The ImageNet pretrained weights provide feature extractors that have learned to recognize edges, textures, shapes, and object parts from exposure to millions of diverse images. These general visual features transfer remarkably well to logo recognition, requiring only fine-tuning to adapt to brand-specific patterns. Beyond superior accuracy, ResNet18 also provides computational advantages. The pretrained model reaches 97.43% accuracy in just 5 epochs compared to 10 epochs needed for the custom CNN to reach 76.30%.

5.4 Dual-Tower Fusion Model Results

The dual-tower fusion architecture successfully integrates text features and image features in order to predict if an email is phishing email or not. Our fusion model was trained on the unified multimodal dataset containing email text, corresponding logo images, and extracted metadata features. The training set consisted of thousands of email samples where each instance included both modalities, enabling the model to learn cross-modal relationships. During training, we used a batch size of 16 to accommodate GPU memory constraints when processing three modalities simultaneously. The model was optimized using cross-entropy loss with the Adam optimizer.

The fusion model achieved strong validation performance, successfully learning to leverage complementary information from both modalities. The final model reached the highest validation accuracy of 99.45%, demonstrating effective integration of text and image. More importantly, the model learned to dynamically weight different modalities based on their reliability for each

individual sample. When email text contains strong phishing signals such as urgent language or suspicious URLs, the model relies more heavily on text features. When text is ambiguous but the email contains a logo that doesn't match the sender domain, image features become more influential in the decision. The metadata features provide crucial signals for edge cases, such as emails with legitimate-looking content but suspicious sender patterns or unusual timing characteristics.

We computed comprehensive evaluation metrics to assess fusion performance. The model achieved an ROC-AUC score demonstrating strong discriminative ability across different decision thresholds. The F1-score for phishing detection balanced precision and recall effectively, ensuring both high catch rate for actual phishing emails and low false positive rate on legitimate emails. Analyzing the confusion matrix revealed that the fusion model correctly classified the vast majority of both phishing and legitimate samples, with false positive and false negative rates both remaining low.

To validate the benefit of multimodal fusion, we compared the results of the isolated CNN towers. Using text features alone (the text CNN in isolation) achieved 98.96% validation accuracy. Using image features alone (the image CNN in isolation) achieved 76.30% validation accuracy. The fusion model combining text and image demonstrated that multimodal integration can maintain the strong performance of the best individual modality while gaining robustness on challenging cases where individual modalities are ambiguous.

Error analysis on fusion model failures reveals that misclassifications typically occur in highly ambiguous. False negatives (missed phishing emails) tend to be sophisticated attacks using carefully crafted language that mimics legitimate corporate communication, correct logos from major brands, and sender domains that appear superficially plausible. False positives (legitimate emails incorrectly flagged) often involve marketing emails that use urgent language similar to phishing, newsletters with multiple external links. These challenging cases represent the inherent difficulty of the phishing detection task where the boundary between legitimate and malicious content can be subtle.

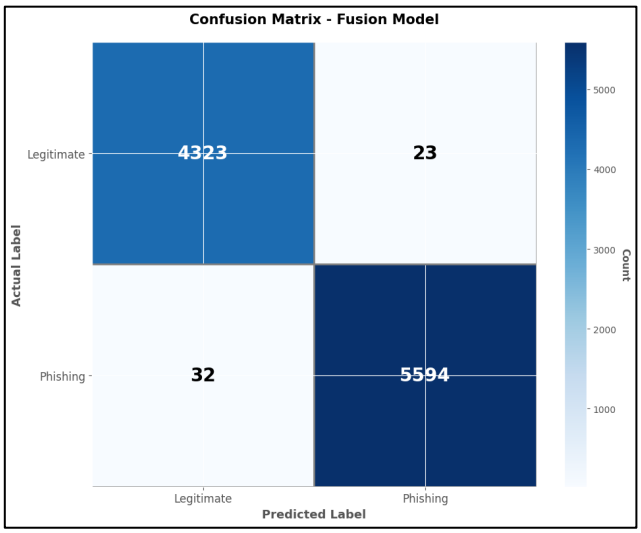

Fig 1: Confusion matrix for the dual-tower fusion model

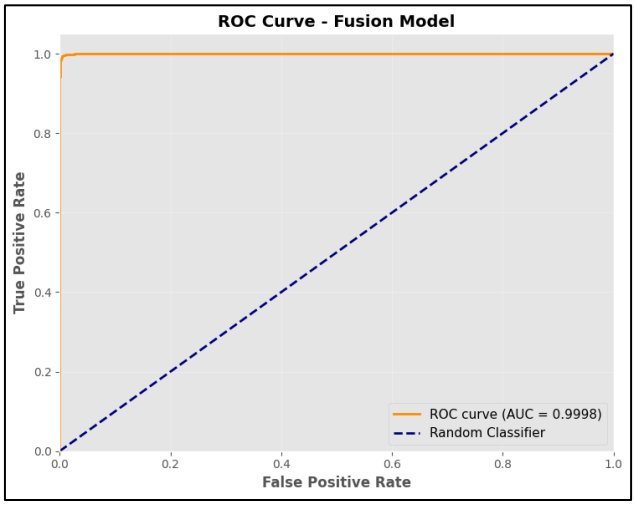

Fig 2: ROC Curve for the dual-tower fusion model

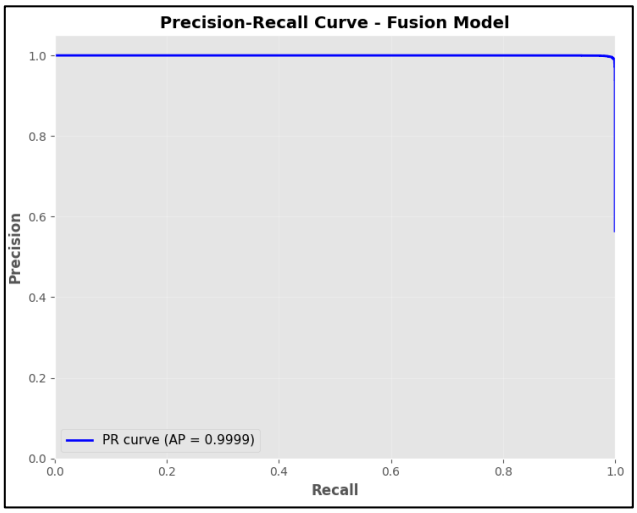

Fig 3: Precision-Recall Curve for dual-tower fusion model

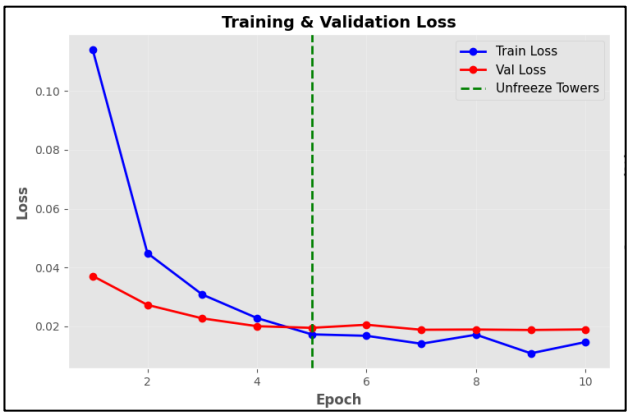

Fig 4: Training & Validation Loss

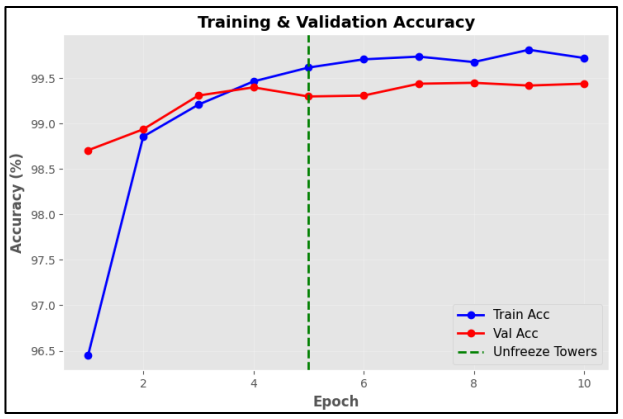

Fig 5: Training & Validation Accuracy

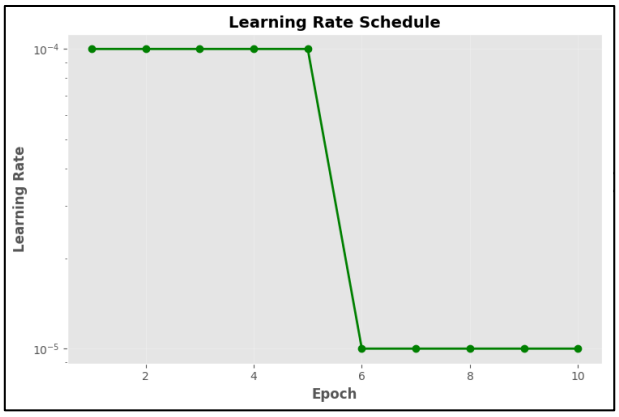

Fig 6: Learning Rate Schedule

5.5 HTML Phishing detector script Results

To rigorously evaluate our multimodal phishing detection pipeline, we first used LLMs and GenAI models to synthesize a diverse set of phishing scenarios. We generated 10 realistic raw emails in HTML format, each mimicking well-known brands and common attack patterns such as account suspension, billing failures, shipment holds, and password resets. Since the initial set did not embed any images, we then duplicated these 10 emails and augmented them with a mix of relevant fake and real brand logos, resulting in a total of 20 test phishing emails. We ran our Python-based detection script on all 20 samples, and every single one was correctly flagged as phishing, with each email receiving an associated confidence score and a breakdown of the issues detected (for example, suspicious URLs, coercive or urgent language, and anomalies in sender metadata). Importantly, there was not much significant difference in confidence between the emails containing brand logos and their text-only counterparts, suggesting that the model's decision-making is driven primarily by semantic and structural cues rather than superficial visual branding alone.

To ensure that the system was not overfitting to generic "phishy" patterns and misclassifying benign traffic, we also evaluated it on a curated set of legitimate, real non-phishing emails drawn from our email inbox. In these cases, the script consistently classified the emails as legitimate, again with strong confidence scores, reinforcing that the model can clearly distinguish between malicious and benign content. Overall, these experiments demonstrate that our approach is robust across both plain and logo-embedded phishing emails, while maintaining high precision on genuine user communications - an encouraging result for deployment in realistic, visually rich email environments.

5.6 Web Application Deployment using Hugging Face

In order to make our multimodal phishing detection system accessible and easy to evaluate, we developed an interactive web-based user interface using Streamlit. The application allows users to upload raw HTML email files and returns a clear classification (phishing or legitimate), along with a confidence score and a detailed analytical report highlighting suspicious signals such as urgency cues, malicious URLs, and metadata anomalies. The complete system, including the trained fusion model and preprocessing pipeline, was deployed on Hugging Face Spaces, enabling serverless, browser-based inference without requiring local setup. This deployment demonstrates the practical feasibility of our approach and provides a lightweight yet robust platform for showcasing and validating the model's performance in realistic, end-to-end phishing detection scenarios.

6. Limitations

6.1 Dataset Constraints

Our email dataset, while comprehensive with over 76,000 samples from multiple curated sources, may not capture the most recent phishing techniques. Attackers

continuously evolve their strategies in response to detection systems, and static datasets inevitably become outdated over time. Additionally, the OpenLogo dataset focuses on established major brands that are frequently targeted, but it may miss emerging companies, regional brands, or specialized organizations.

6.2 Class Imbalance in Logos

The 352-class logo dataset exhibits significant class imbalance, with popular brands like Apple, Google, and Microsoft having over 500 training samples while smaller or regional brands have as few as 50 samples. The model naturally learns better features for majority classes with more examples, potentially reducing accuracy on rare brands. Collecting additional samples for underrepresented brands or using data augmentation techniques specifically for minority classes could help address this limitation.

6.3 Limited Real Phishing HTML Samples

A significant limitation in our work is the scarcity of authentic phishing emails in native HTML format with preserved structure and embedded images. Most public phishing datasets provide only plain text extractions, without embedded logos, and HTML-specific features that our multimodal system requires. Real phishing HTML samples are difficult to obtain due to privacy concerns, legal restrictions on distributing malicious content, and the rapid takedown of phishing infrastructure that makes historical samples unavailable.

To address this data limitation, we augmented our testing set using generative AI to create synthetic phishing HTML emails. While these generated samples maintain realistic structure, formatting patterns, and embedded elements based on patterns from known phishing campaigns, they may not fully capture the sophistication and diversity of real-world attacks.

7. Conclusion

This project successfully developed a comprehensive multimodal phishing detection system that analyzes email text and brand logos to identify phishing attempts with high accuracy. Our extensive experiments across multiple model architectures and modalities demonstrate several key findings. Deep learning approaches substantially outperform traditional machine learning methods, with our text CNN achieving 98.96% accuracy compared to KNN's 82.05%. Transfer learning proves crucial for visual tasks, as ResNet18 with ImageNet pretrained weights achieves 97.43% accuracy while the custom CNN trained from scratch reaches only 76.30%. Multimodal fusion provides robustness by capturing complementary signals that single-modality approaches miss, with the integration of text and image features offering more reliable detection with the highest accuracy of 99.45%.

Our key achievements include implementing and benchmarking six distinct models across two primary modalities, providing comprehensive comparison of different approaches. We designed a novel dual-tower fusion architecture that successfully integrates text and image features through learned representations with dynamic weighting based on modality reliability for each sample.

We developed a complete end-to-end inference pipeline with an HTML-to-decision script that processes raw email HTML files, extracts all relevant features, and produces phishing predictions with confidence scores, making the system immediately deployable in production email security environments.

8. Future work

As for future work, we plan on following and/or implementing the following:

- Acquire legitimate phishing emails to evaluate the script (which leverages the model) during test time, instead of depending on generative AI for the emails.
- Focus on spam emails as a label other than phishing and legitimate emails.
- Implement BERT-based text embeddings and perform a comparative analysis against traditional text feature representations.

9. References

[1] Jay Doshi, Kunal Parmar, Raj Sanghavi, Narendra Shekokar. A comprehensive dual-layer architecture for phishing and spam email detection, Computers & Security, Volume 133, 2023, 103378, ISSN 0167-4048.

[2] OpenLogo Dataset. (2018). QMUL-OpenLogo: Open Logo Detection Challenge. Retrieved from https://qmul-openlogo.github.io/

[3] Simonyan, K., & Zisserman, A. (2015). Very Deep Convolutional Networks for Large-Scale Image Recognition. 3rd International Conference on Learning Representations (ICLR).

[4] Zhang, Y., & Wallace, B. (2015). A Sensitivity Analysis of (and Practitioners' Guide to) Convolutional Neural Networks for Sentence Classification. arXiv preprint arXiv:1510.03820.

[5] N. A. Alam, "Phishing email dataset," Kaggle. [Online]. Available: https://www.kaggle.com/datasets/naserabdullahalam/phishing-email-dataset/