Project Link: <https://drive.google.com/drive/folders/1lODNiW6hGUa2kBORmOgnSjIEv8roSNe1?usp=sharing>

* In the website folder we have the index file so the file name “fake” need to be in htdoc folder of xampp directory.
* Put the model and all the “website” folder files in one folder for testing the website.
* Then launch the website api Using terminal bash - “python api.py”

Name Student ID Contribution

Jeshfique Mahmud (Model training) 2111097642

- Designed the neural network architecture using ‘ResNeXt-50’

-Designed the Bert architecture using “Bert model”

- Designed the fusion model for ResneXt 50 and Bert

-Implement the enhancedBert and the fusion model

-Designed the API for website

- Implemented the ModifiedResNeXt class with custom classifier layers

- Led model training and validation

Full model training and evaluation for all three models:

– BERT Text Classifier

– ResNeXt-50 Image Classifier

– Fusion Model (BERT + ResNeXt)

- Hyperparameter tuning

- Model checkpointing and result saving

Syeda Nusaiba Sharifeen (*Preprocessing Lead)* 2132576642

- Handled \*dataset cleaning, splitting, and ‘custom DataLoader’ creation

- Applied \*advanced image augmentation techniques\* for better generalization

- Implemented ‘class balancing’ using WeightedRandomSampler

- Evaluated model with ‘confusion matrix, ‘classification report, and ‘cross-validation’

Image preprocessing:

– Filtering invalid files, resizing, augmentations (rotation, flipping, color jitter)

🔹 Dataset balancing

🔹 Train/Val/Test split setup for images

🔹 Custom transform pipelines

* Website building for running the model testing.

Chadne. *Preprocessing Lead)* 2132112464

- Documented the methodology, training results, and augmentation strategies

- Designed the \*block diagram\* representing the model's architecture

- Created inference scripts for real-world testing of trained models

- Compiled \*result tables\*, training logs, and final test accuracy summary

| 🔹 Text preprocessing:  – Cleaning, tokenization, padding, and augmentation  🔹 SMOTE and label encoding  🔹 Dataset splitting (train/val/test)  🔹 Torch Dataset/DataLoader setup for text   * Website building for running the model testing * Testing for the models were also done. |
| --- |

Model Architecture and their Working Principles:

1. Text Model: Enhanced BERT Classifier : In the Enhanced BERT classifier, a BERT-based-ucased backbone architecture is used followed by a dropout layer that helps reduce overfitting. A fully connected denslayer is used for the classification. Cleaned and tokenized news article text is used as the input. This input is then converted into token embedding using BERT. The contextual understanding is developed using the transformer layers. The [CLS] token is used for classification. The dense output later makes the final prediction between real or fake. The loss function is determined by the CrossEntropyLoss or the Focal Loss. A probability distribution is received between the real and the fake class. The transformer layers provide a strong contextual reasoning to the data. However, the visual cues received are crucial and any incorrect or misleading information may lead to vast errors.
2. Image Model: Improved ResNeXt-50: The ResNeXt-50 architecture is a pretrained model on ImageNet. It is modified by adding batch normalization and dropout as well as replaceung final classifier layer for binary classification. A resized and augmented image of the news post is used as input. The images go through a convolutional and grouped convolutions(ResNeXt block). The feature maps are downsampled and encoded. The final classification is done by using a softmax function. The Loss Function is evaluated by the CrossEntropyLoss. The output here is given based on visual patterns. This model detects visual cues like manipulated or irrelevant photos. However, it ignores textual context which can lead to misinterpretation caused by good-looking fake visuals.

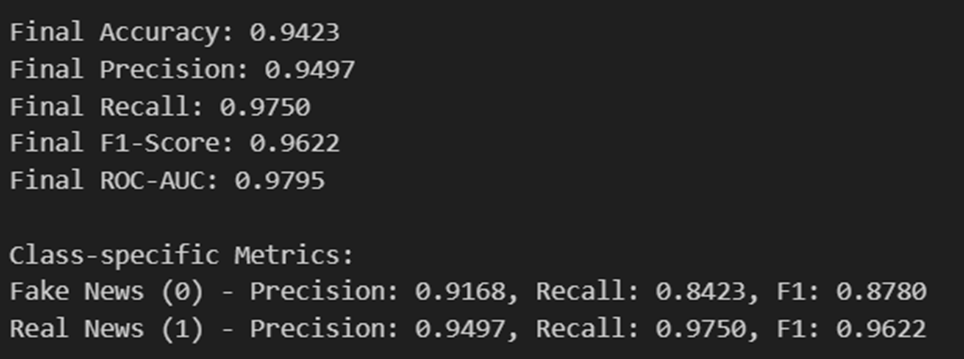
1. Fusion Model: Multimodal Fake News Detector(BERT + ResNetXt): The image model and the text model both run parallelly. Outputs from both the branches are concatenated and pass through a fully connected fusion layer which is the binary layer. Both the text and the corresponding image are taken as input here. The text branch encodes the semantic meaning of the article and the image branch captures the visual features. The fusion layer combines both the modalities for a joint reasoning. The Loss function is evaluated using CrossEntropyLoss. The output gives a prediction combining the results from both the text and the image. This provides a balanced decision making. However, that requires well-aligned text-image pairs and the computation is expensive.

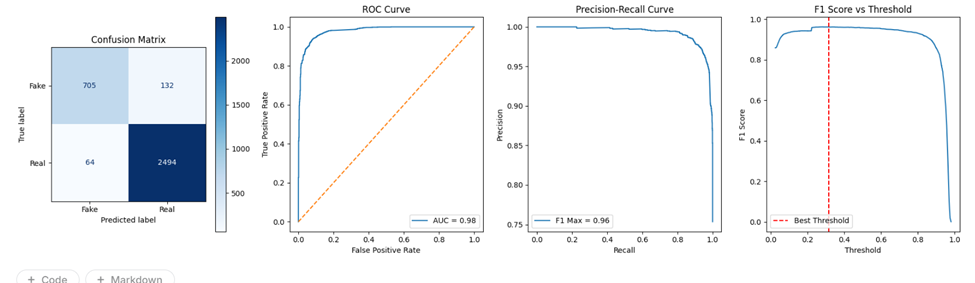
General Pipeline Summary:

As we have both text type data and image type data, the data is initially preprocessed separately. The text is all converted to lowercase, cleaned and tokenized using the BERT tokenizer. The images are resized, normalized and then some heavy augmentation is done. For the training setup, the AdamW optimizer is used and the linear decay with warmup is used for the scheduler. The results are evaluated based on the accuracy, precision, recall and the F1-score.

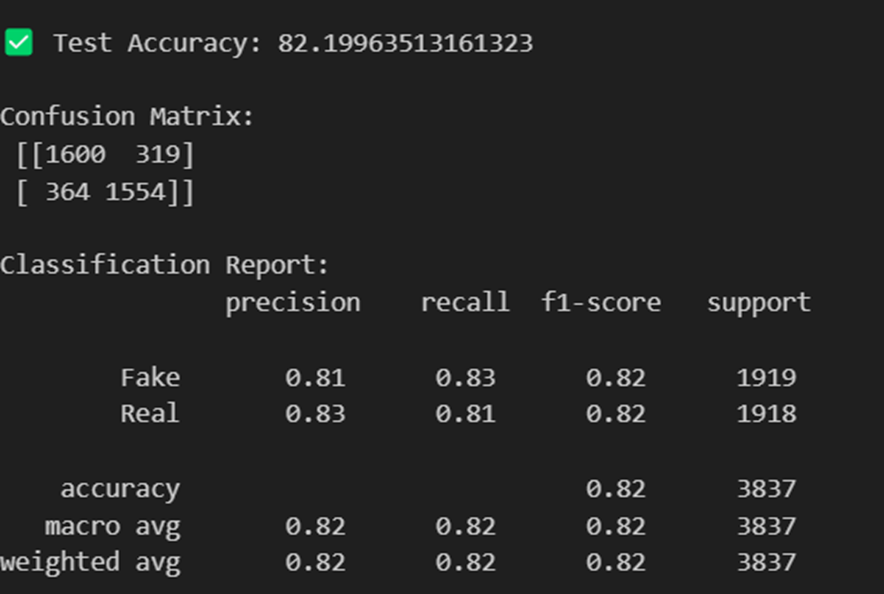
**Confusion Matrix and Results:**

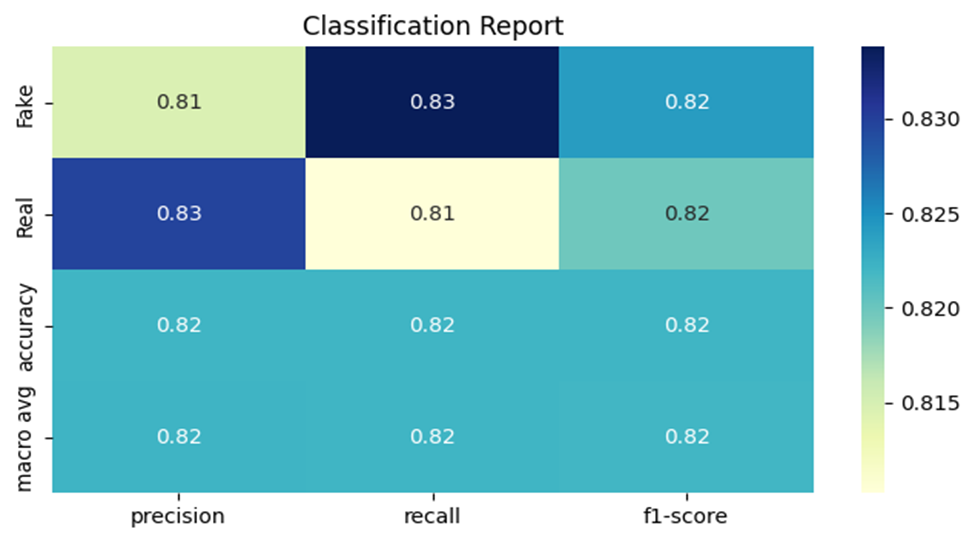
**For Text:**

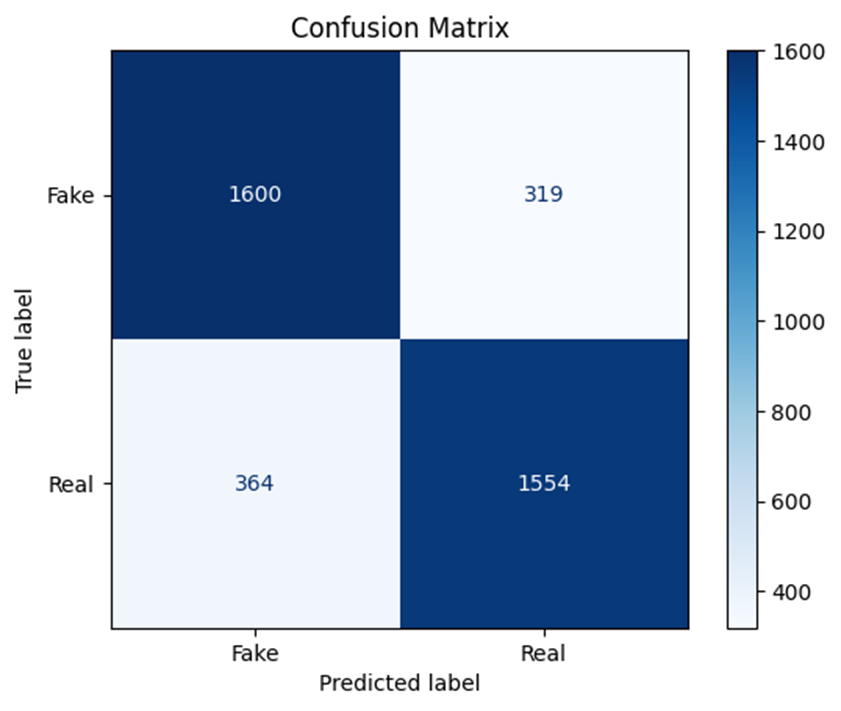
****

****

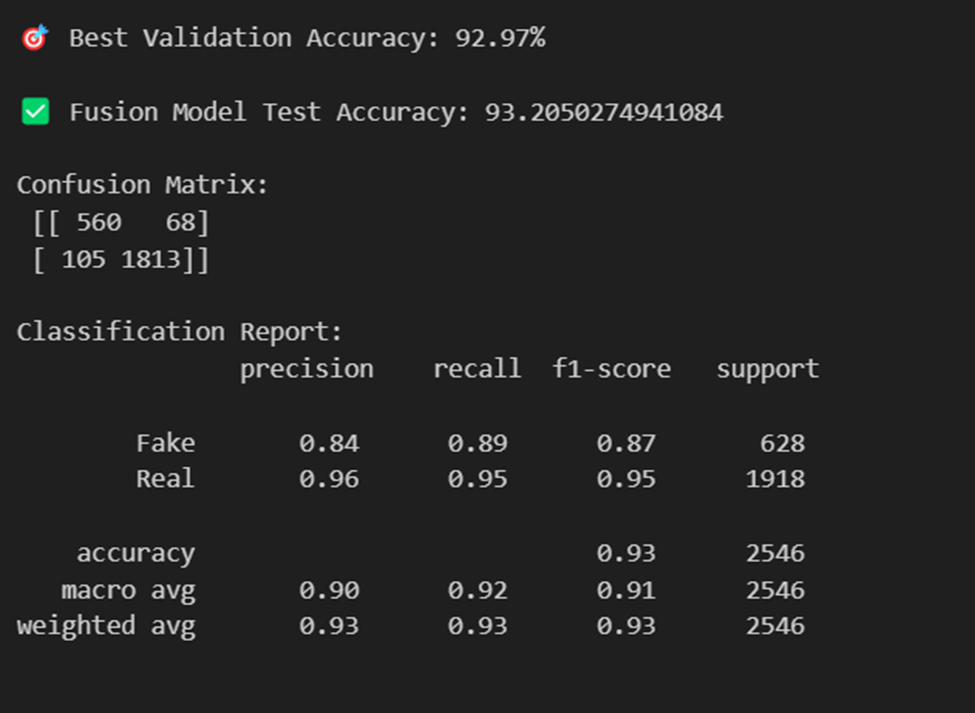
**For Image:**

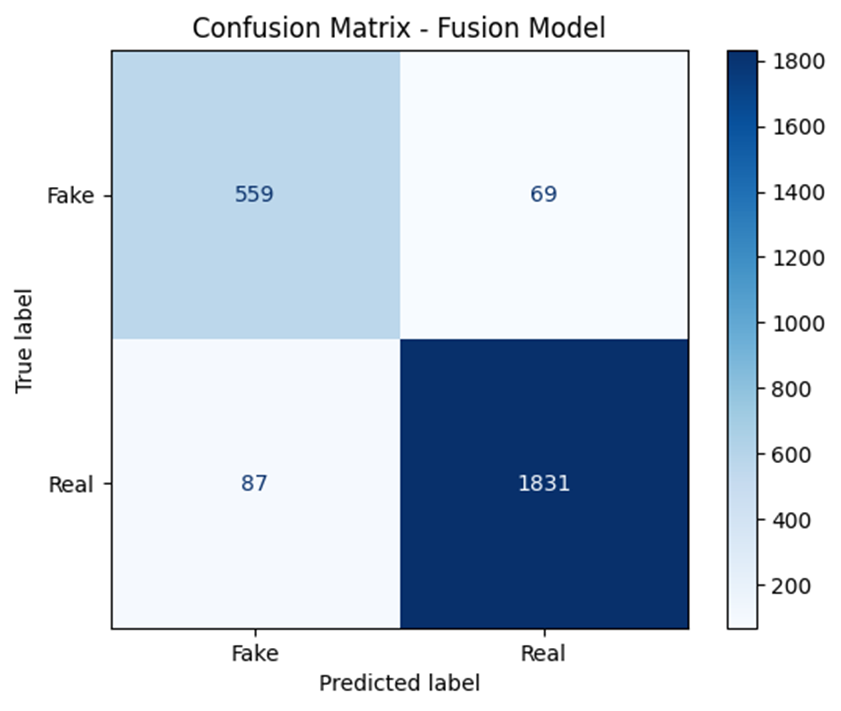
****

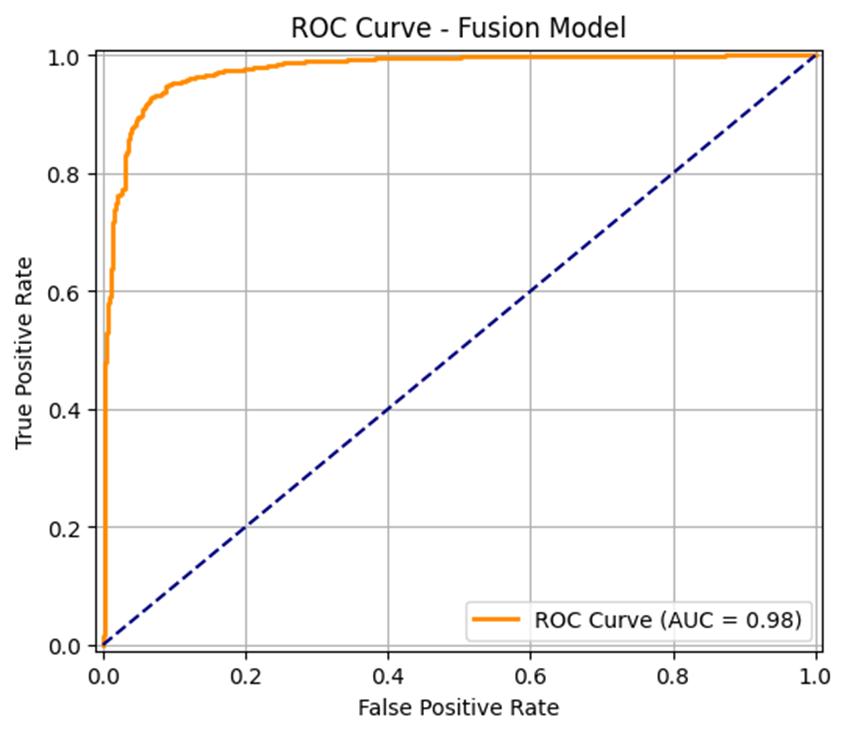
****

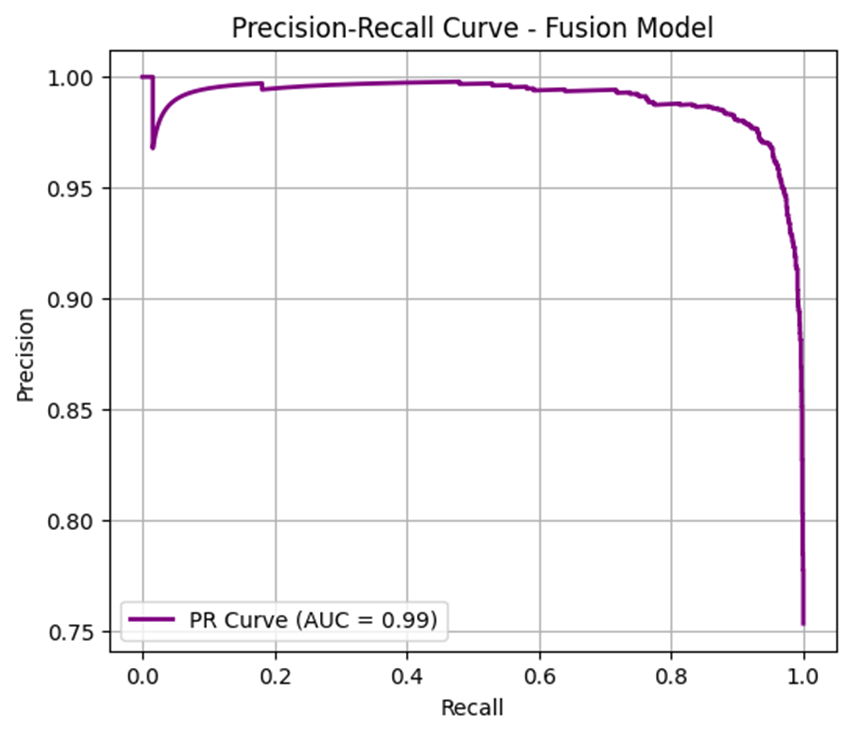


**For Fusion Model:**

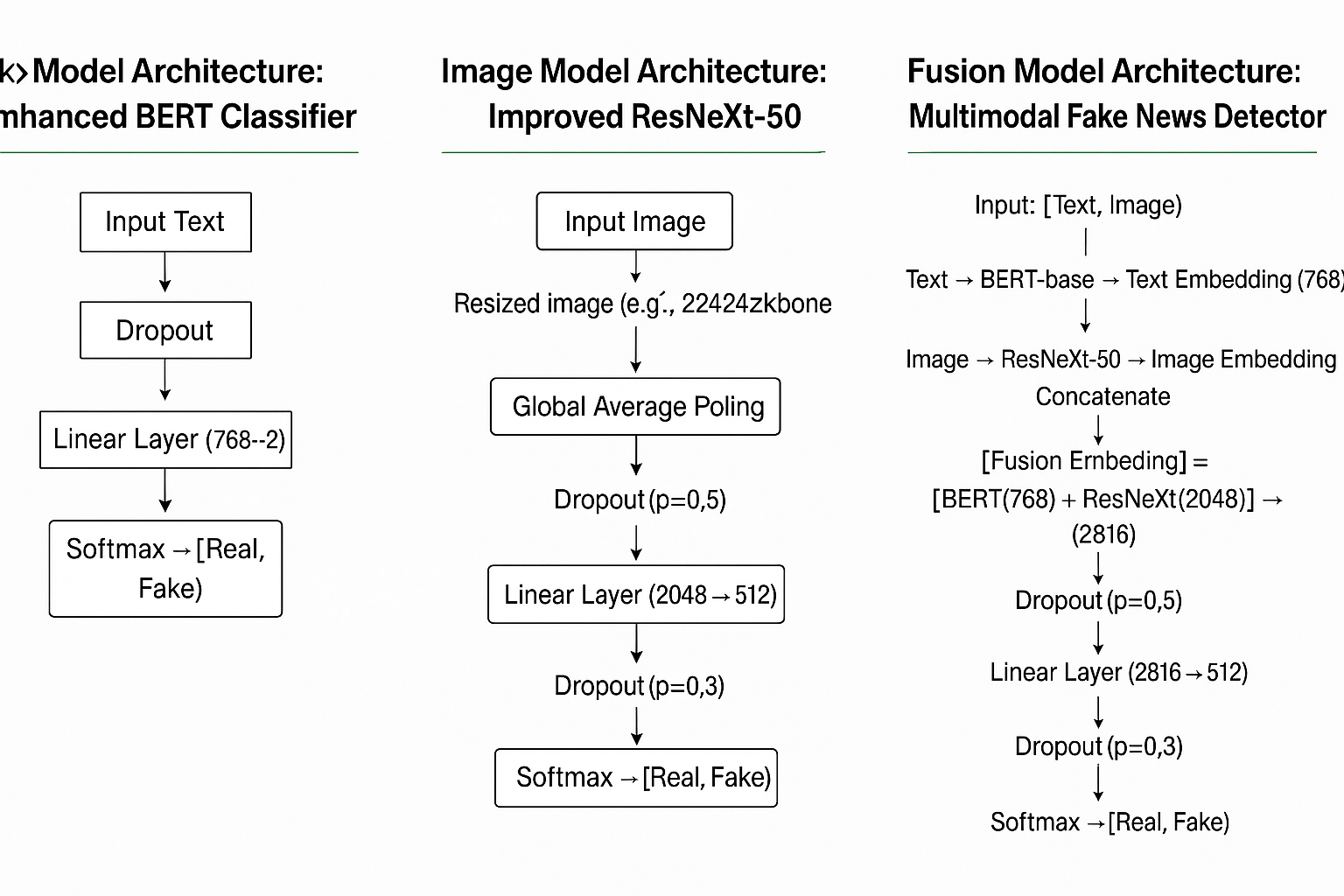
****

****

****

****

**Network Architecture:**

****