MULTI-MODAL GRAPH TRANSFORMER

ES667 - DEEP LEARNING PROJECT

Bhavya Parmar - 23110059
Rishabh Jogani - 23110276
Umang Shikarvar - 23110301

PROBLEM STATEMENT

- **Task:** Multimodal Sentiment Analysis using Graph Transformer – Predict sentiment from image and text pairs.

- **Objective:** Leverage graph transformers to better model structural dependencies between image-text pairs and outperform standard baselines.

- **Good Model:** One that effectively fuses visual and textual modalities and generalizes well to unseen sentiment cases.

DATASET USED

- Twitter Dataset for Sentiment Analysis
- **4869** Samples of image and text data (Train/val/test split = 80/10/10%)
- **Image shape:** (256, 256, 3)
- **Text:** A caption for each image
- **Sentiment classes** – Positive (2) / Neutral (1) / Negative (0)

ARCHITECTURES

Architecture 1:

1. Get Embeddings for Image as well as Text data
2. Concatenate the two and pass to a single transformer
3. Pass the output from the transformer to the MLP layer

Architecture 2:

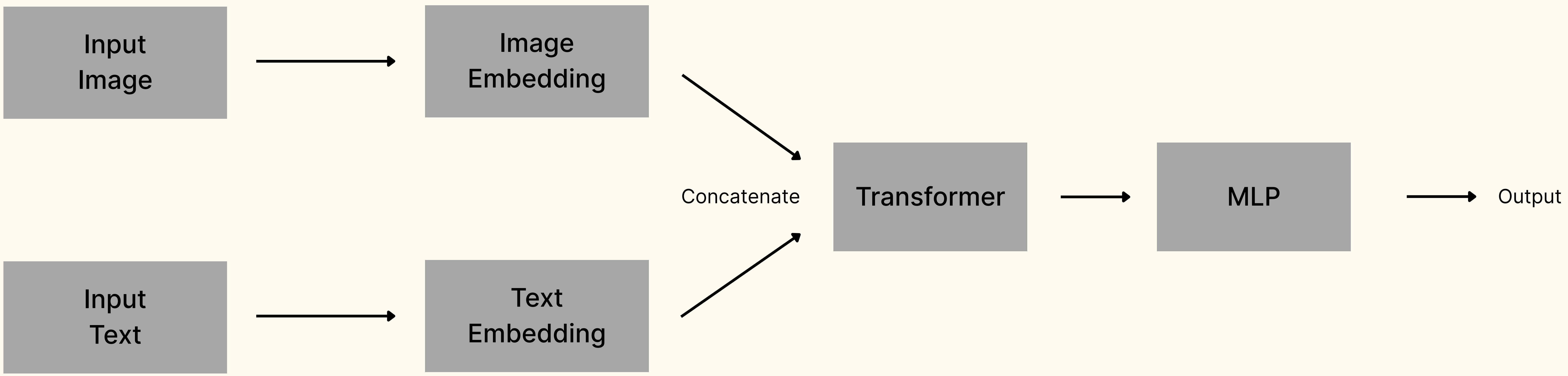
1. Get Embeddings for Image as well as Text data
2. Pass it through separate Transformers
3. Concatenate the two and pass to the MLP layer

ARCHITECTURES

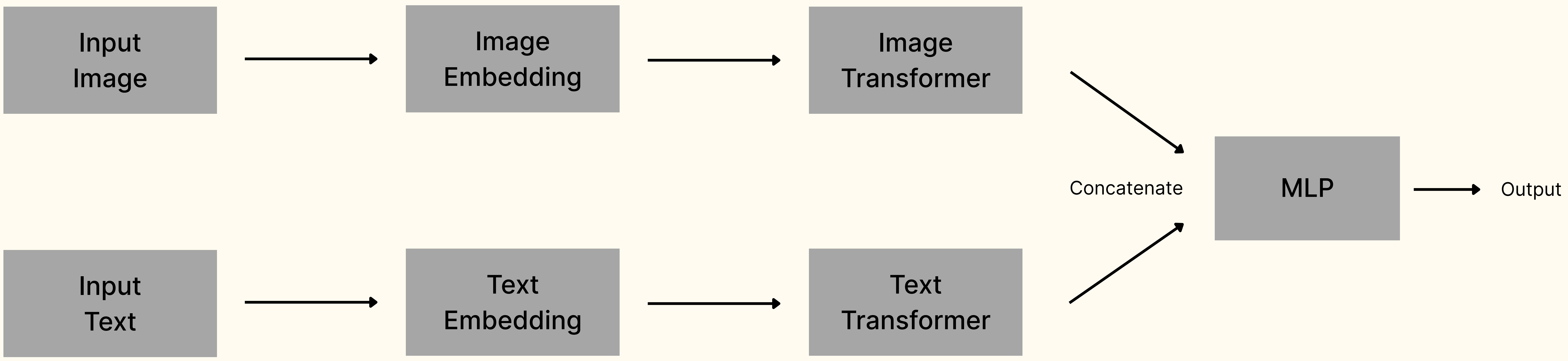
Architecture 2 was implemented in four different configurations:

1. Graph Transformers for both image and text.
2. Graph Transformer for the image and a regular Transformer with Sinusoidal-Cosine positional encoding for the text.
3. Regular Transformers for both image and text, without positional encoding.
4. Regular Transformers for both image and text, with Sinusoidal-Cosine positional encoding.

ARCHITECTURE - 1



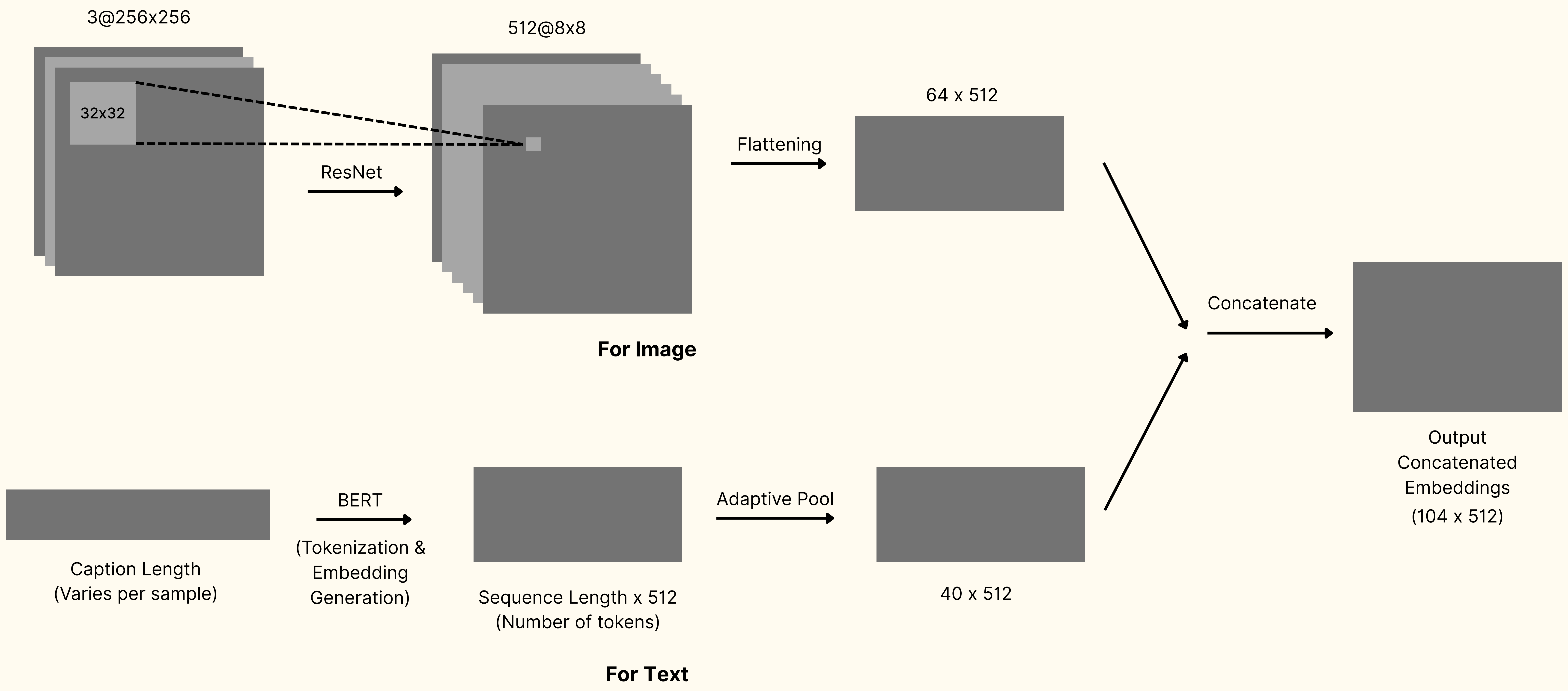
ARCHITECTURE - 2



TRAINING CONFIGURATION

- **Optimizer:** AdamW
- **Learning Rate:** 2e-5 with Cosine Annealing Scheduler
- **Batch Size:** 16
- **Loss**: Cross Entropy Loss
- **Epochs:** 15
- **Evaluation Metric:** Accuracy
- **Checkpointing & Logging:** Best model saved based on validation loss
- **Hardware:** NVIDIA P100 GPU (Kaggle)

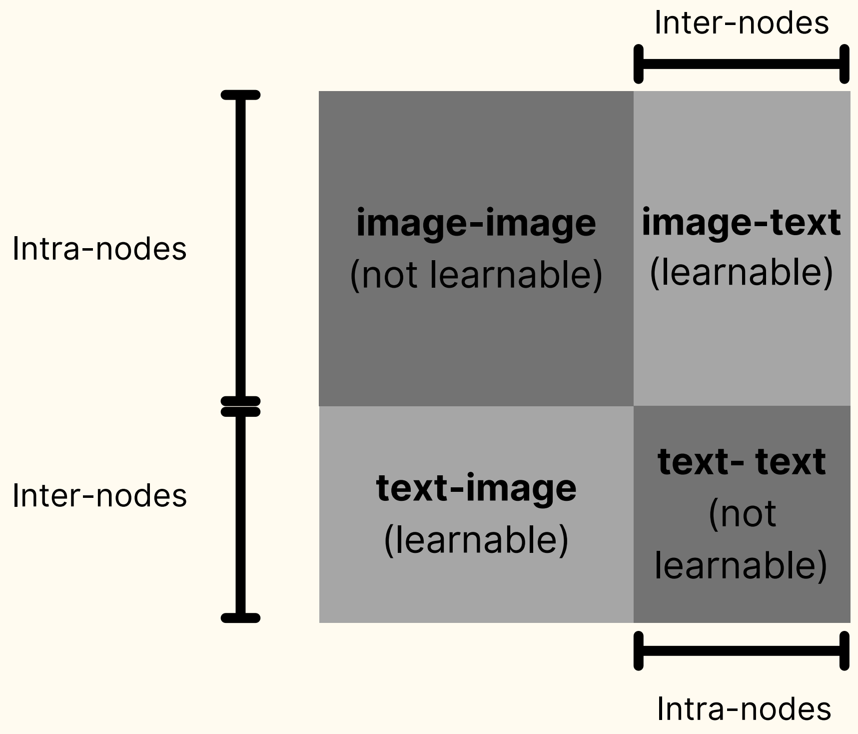
Architecture 1- **Generating Image and Text Embeddings** (Preprocessing)

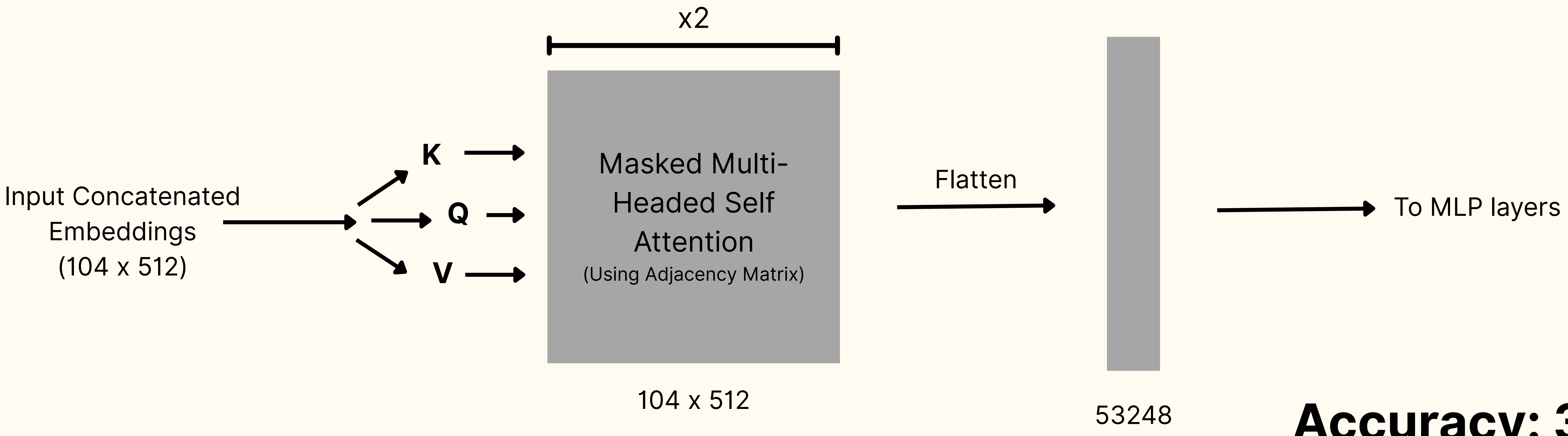


Architecture 1 - **Transformer**

For our Mask (Adjacency Matrix):

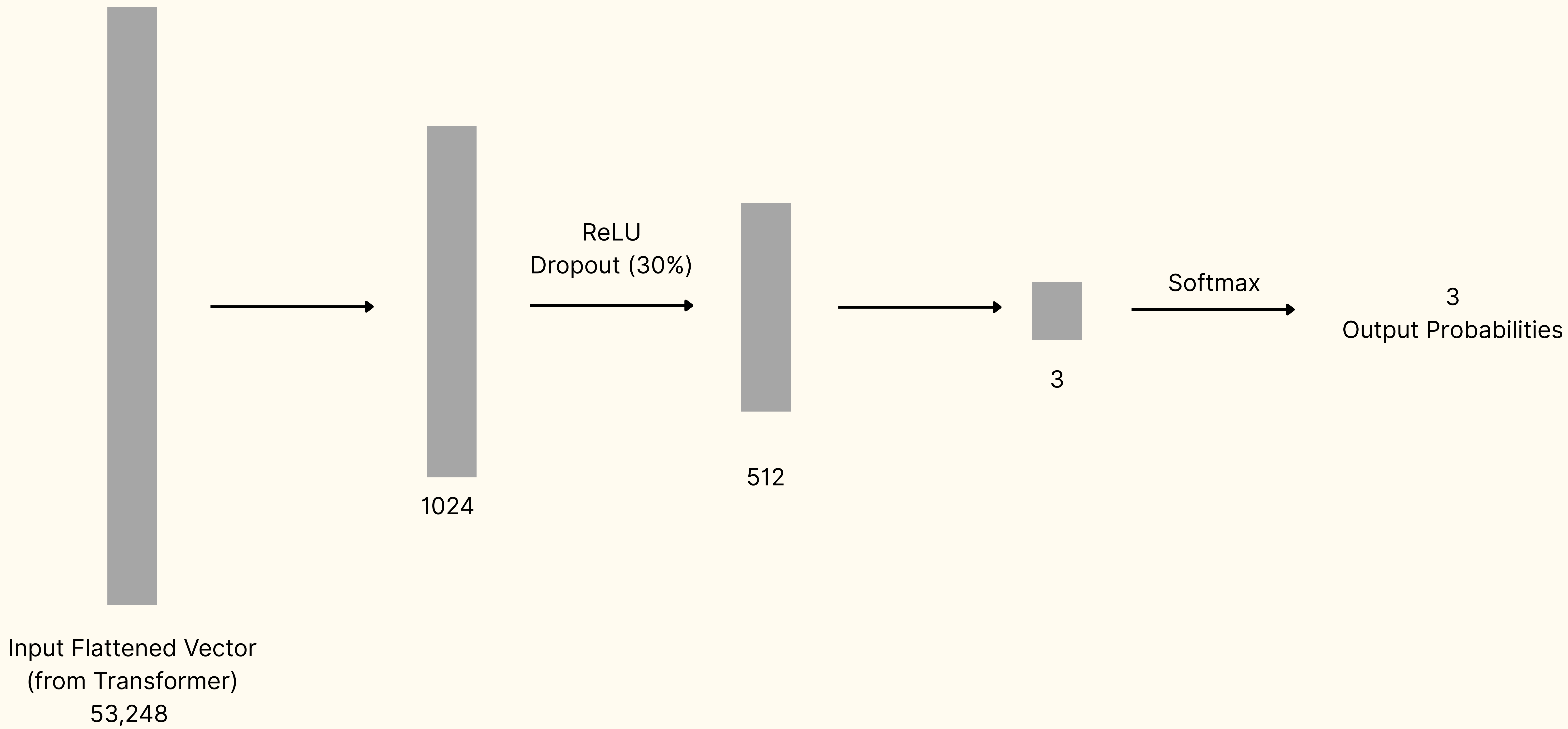
- For the intra-nodes (image-image, text-text) the graph edges in adjacency matrix were fixed (using nearby neighbours relation for image and sliding window method for text)
- For the inter-nodes (image-text, text-image) the graph edges in adjacency matrix were kept learnable (using gradient descent)



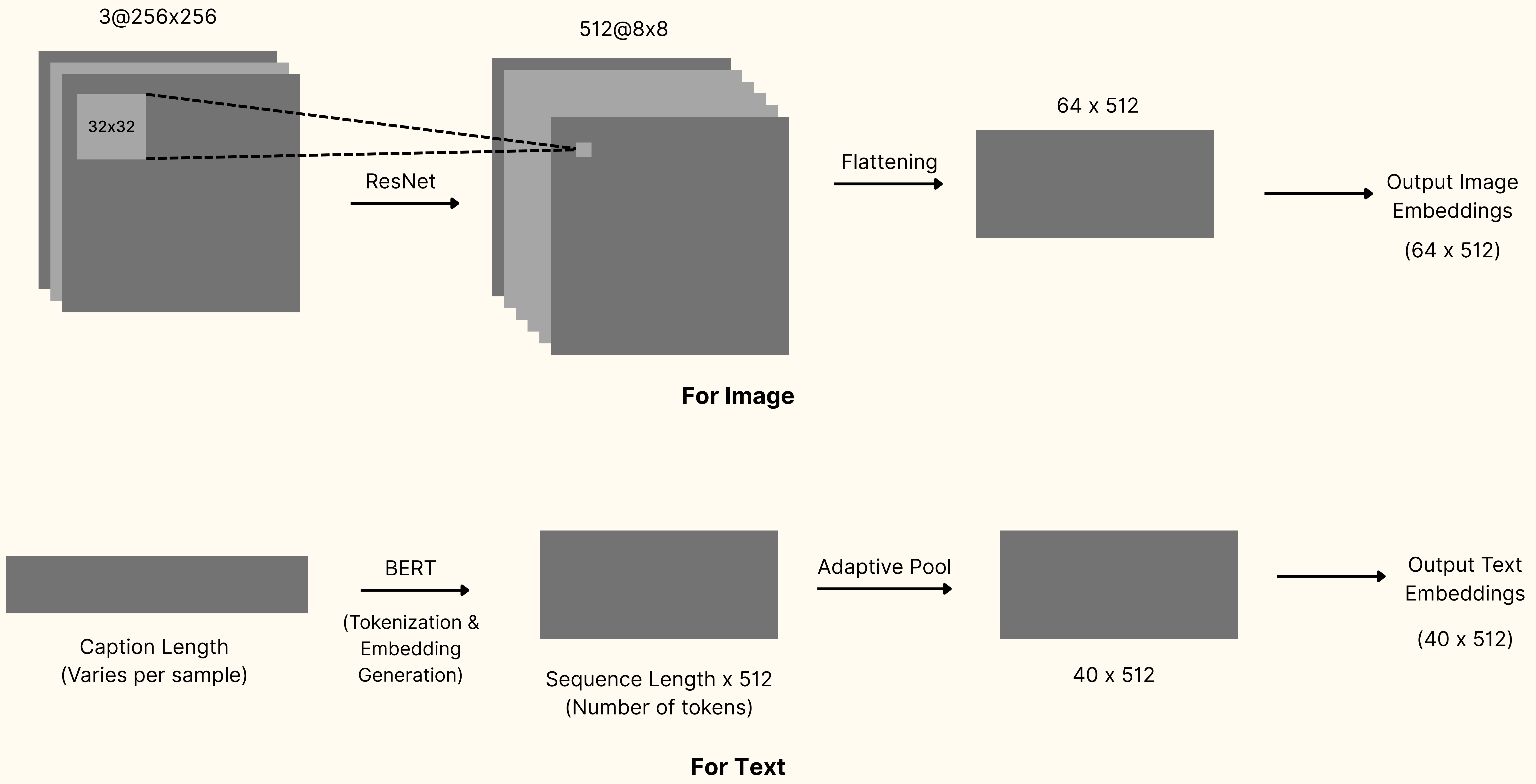


Accuracy: 37.78%

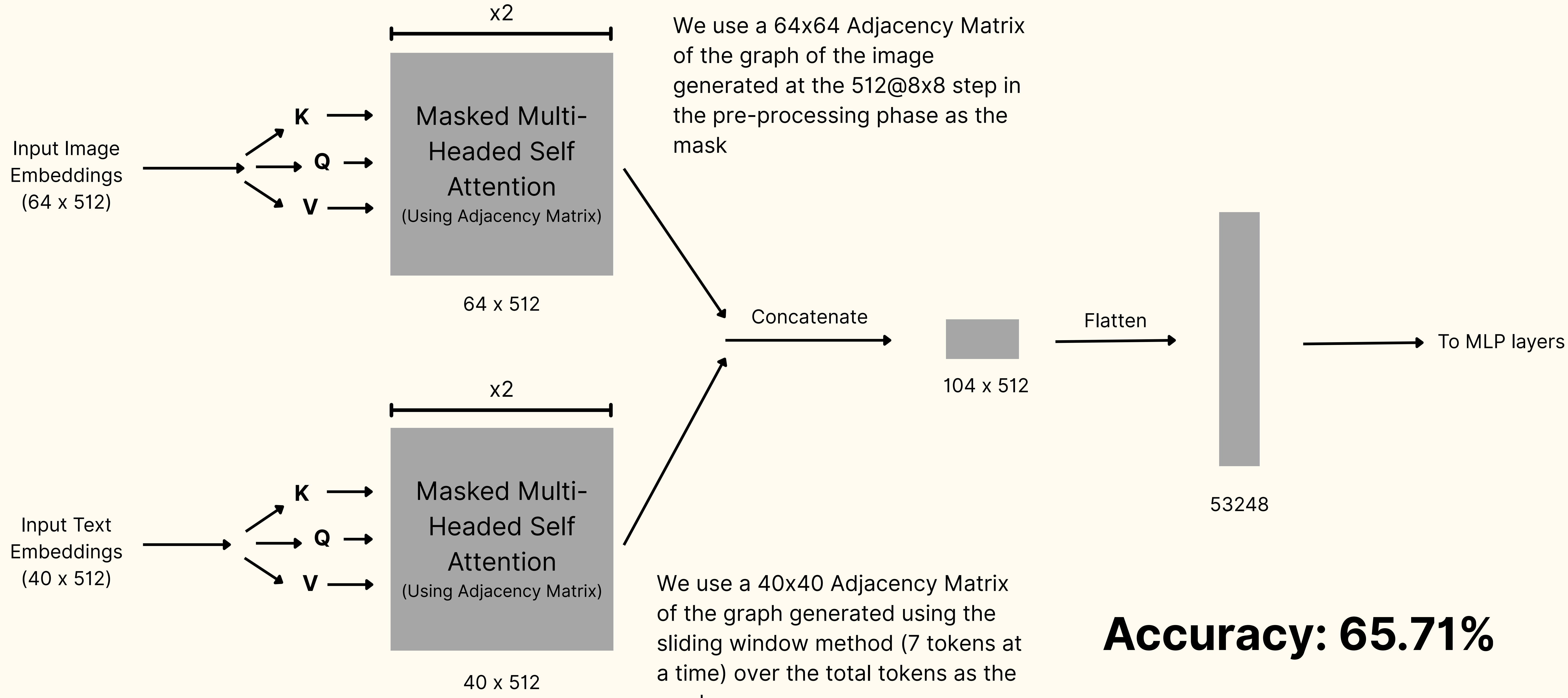
MLP Layers


ReLU
Dropout (30%)
Softmax
3
Output Probabilities
3
512
1024
Input Flattened Vector
(from Transformer)
53,248

Architecture 2- **Generating Image and Text Embeddings** (Preprocessing)



Method 1 - **Using Graph Transformer for both Image and Text**

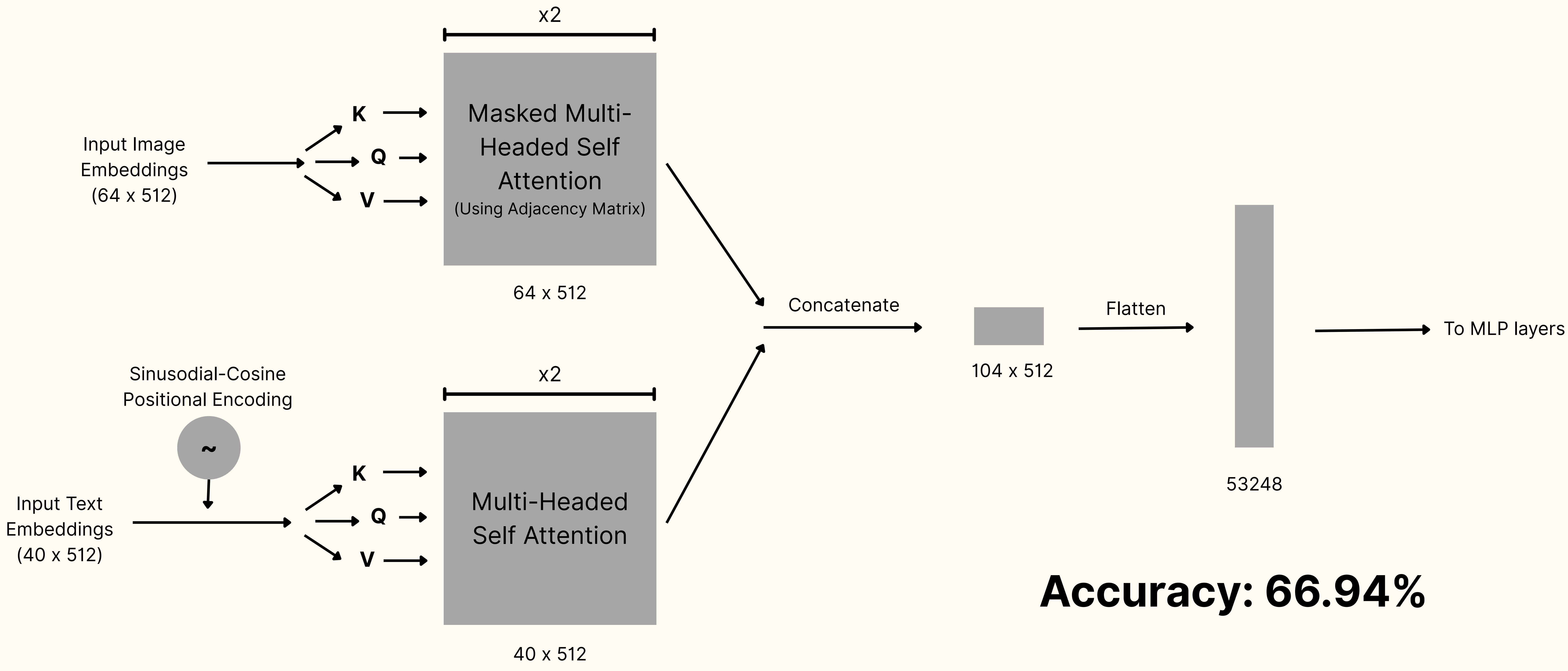


We use a 64x64 Adjacency Matrix of the graph of the image generated at the 512@8x8 step in the pre-processing phase as the mask

We use a 40x40 Adjacency Matrix of the graph generated using the sliding window method (7 tokens at a time) over the total tokens as the mask

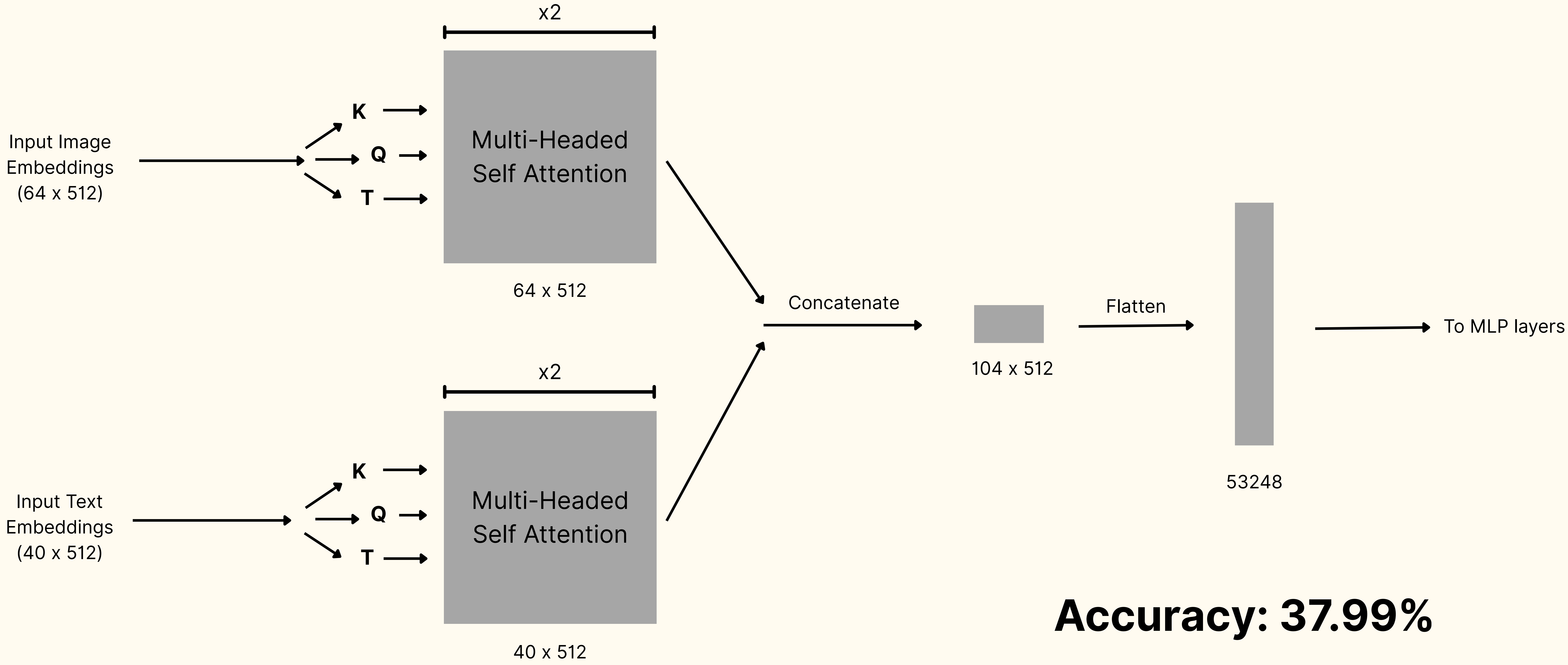
Accuracy: 65.71%

Method 2 - **Graph Transformer for Image and Regular Transformer for Text (with Sinusoidal-Cosine positional encoding)**



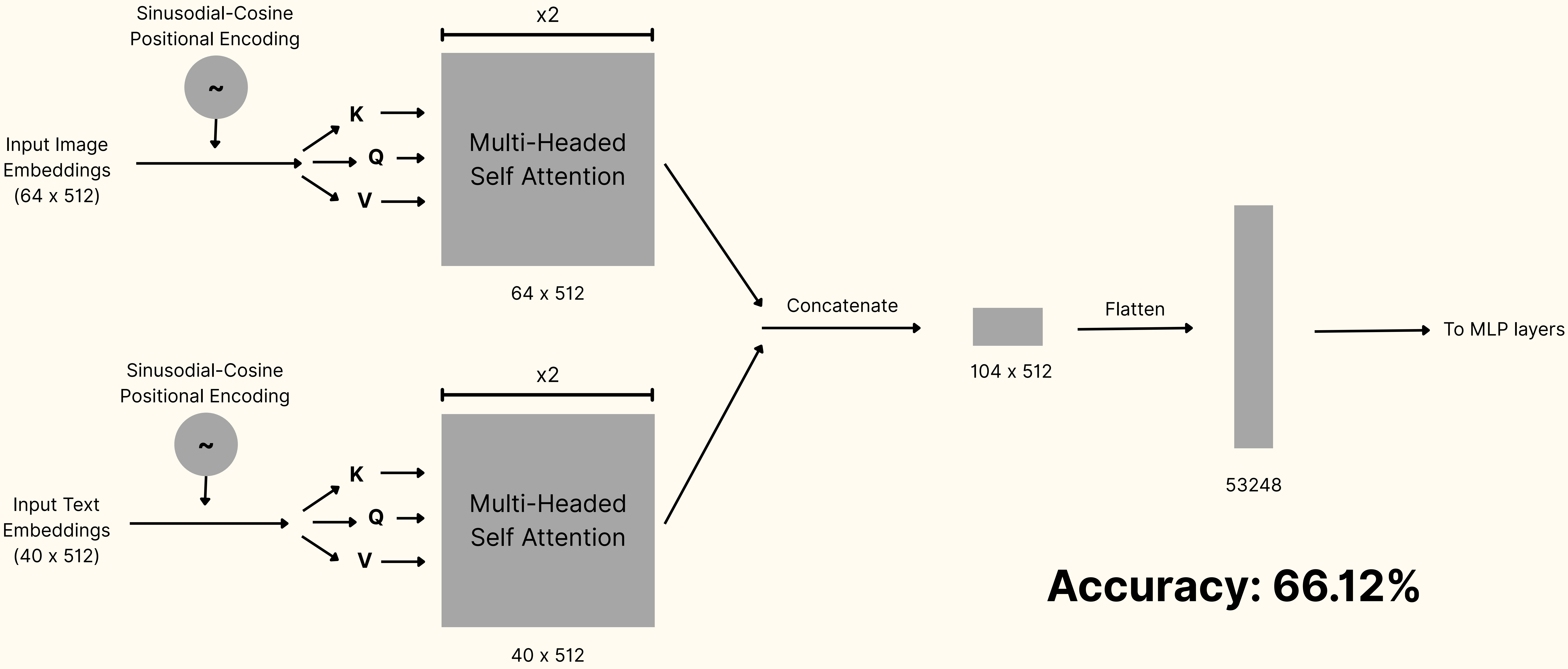
Accuracy: 66.94%

Method 3 - **Using Regular Transformers for both Image and Text without Positional Encoding**



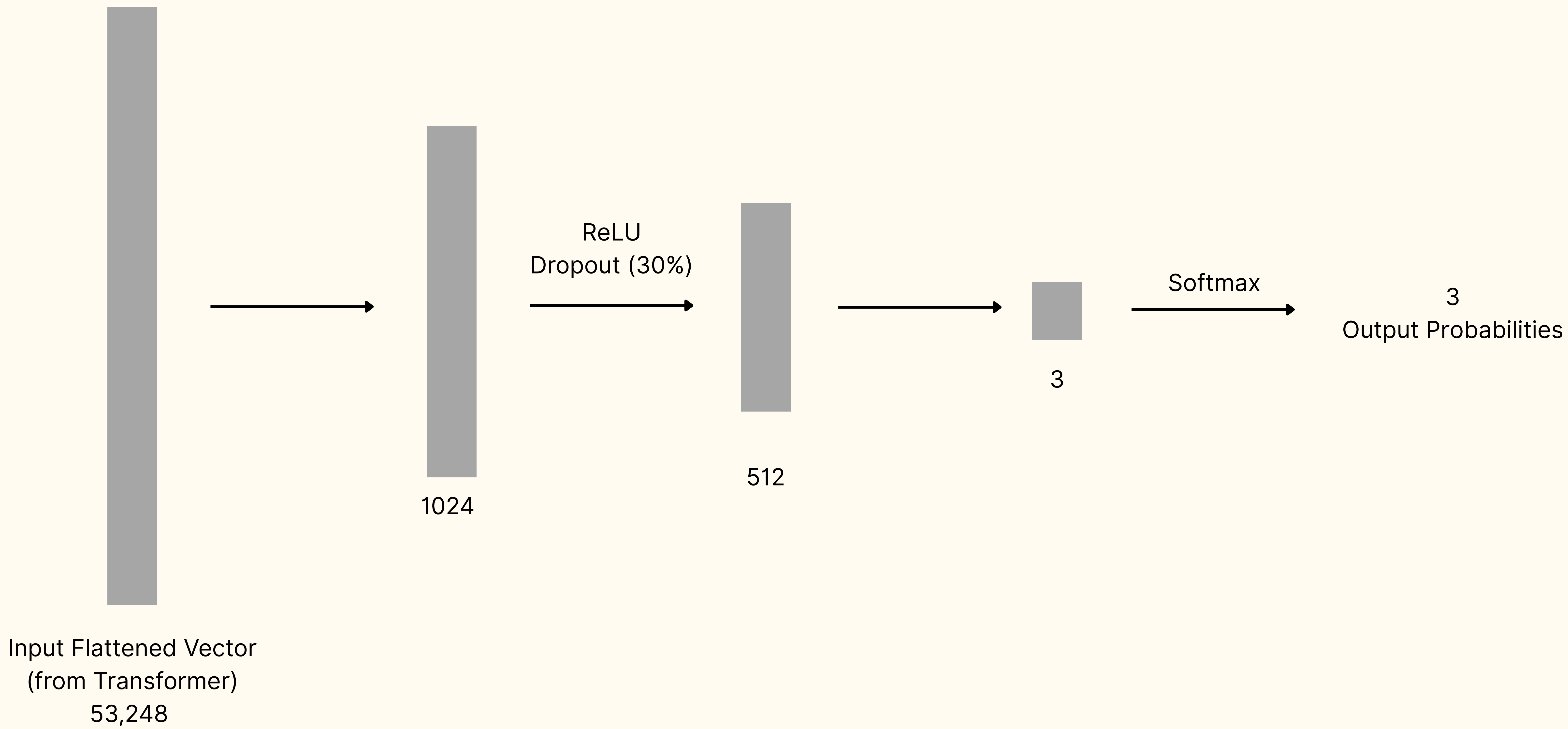
Accuracy: 37.99%

Method 4 - **Using Regular Transformers for both Image and Text with Sinusoidal-Cosine Positional Encoding**



Accuracy: 66.12%

MLP Layers


ReLU
Dropout (30%)
Softmax
3
Output Probabilities
3
512
1024
Input Flattened Vector
(from Transformer)
53,248

RESULTS

	MODEL	ACCURACY
Architecture - 1	Single Graph Transformer	37.78
Architecture - 2	Graph Transformer for both	65.71
	Graph Transformer for Image and regular Transformer for text	66.94
	Regular Transformer for both with p.e.	66.12
	Regular Transformer for both without p.e.	37.99

LIMITATIONS

- The dataset used was largely uncurated, consisting of randomly sampled Twitter images and captions
- Used pre-trained BERT and RESNET models to get encodings for data due to lack of time and compute to get them ourselves.
- The architecture assumes both image and text contribute equally to sentiment — in reality, either modality may dominate or be irrelevant.

SCOPE FOR FUTURE

- Extend beyond image and text by integrating audio (e.g., tone of voice, background sounds) or short video clips, allowing the model to capture richer emotional cues.
- Currently, intra-node (image–image, text–text) graph edges are fixed (nearest neighbours/sliding-window). Making these edges learnable could let the model discover more semantically meaningful graph structures within each modality.
- Introduce mechanisms (e.g., attention-based gates) that allow the model to learn how much each modality should contribute per instance.
- Rather than relying on fixed pre-trained BERT and ResNet embeddings, fine-tune (or train from scratch) the visual and textual encoders on the target sentiment dataset to better adapt to its domain-specific patterns.

CONCLUSION

- Utilising graph transformers for image data proves effective, as it significantly enhances accuracy by capturing spatial relationships between visual elements.

- Applying graph transformers to textual data yields minimal improvement, likely due to the sequential nature of language already being well-modelled by standard transformers.

- Processing different modality embeddings separately at lower levels allows each modality to capture its unique semantic patterns more effectively. By combining the high-level features afterward, the model benefits from richer and more complementary representations, ultimately leading to improved performance in multimodal tasks.

PYTHON NOTEBOOKS

- https://www.kaggle.com/code/umangshikarvar/dl-tranformer/edit
- https://www.kaggle.com/code/umangshikarvar/gtn-2/edit
- https://www.kaggle.com/code/umangshikarvar/dl-gnn-1/edit

Thank you.