# Project Report

## KANVisionLSTM\_FFT: Deepfake Detection and Image Classification

### 1. Introduction

This project introduces a dual-purpose deep learning model designed to detect real vs fake images and simultaneously classify them into one of three categories: human\_faces, animals, or vehicles. The model integrates frequency-domain features using FFT with backbone CNN or ViT architectures and applies sequential reasoning using LSTM with attention mechanisms.

### 2. Methodology

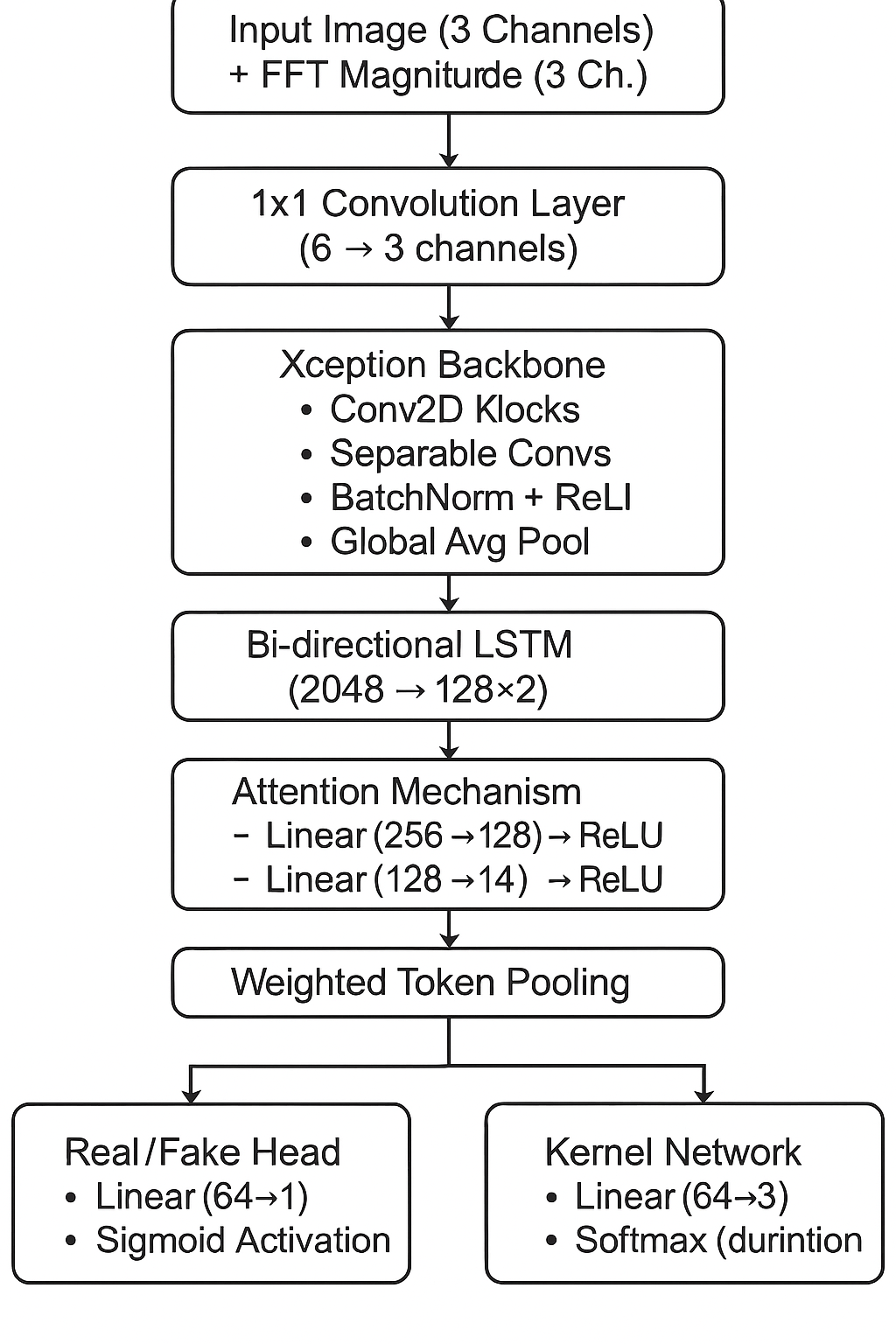
The training pipeline begins by loading and preprocessing the ArtiFact\_240K dataset. Each image is converted to RGB, resized, augmented, and normalized. A custom PyTorch Dataset is used to assign both binary (real/fake) and multi-class labels.  
  
The model architecture is structured as follows:

- Input: RGB image (3 channels) + FFT magnitude (3 channels), concatenated into 6 channels.  
- 1x1 Convolution: Projects 6-channel input back to 3 channels.  
- Backbone: A CNN (Xception by default) or Vision Transformer to extract spatial features.  
- LSTM: Processes the flattened features with a bidirectional LSTM.  
- Attention: Learns importance weights over token sequences.  
- Kernel Network: Applies dense layers for final transformation.  
- Dual Output Heads:  
 - Sigmoid for binary classification (real/fake)  
 - Softmax for class prediction (human\_faces, animals, vehicles)

### 3. Dataset

The ArtiFact\_240K dataset is structured into three subsets: train, validation, and test. Each contains images categorized under real/fake and one of the three object classes. During training, transformations like horizontal flipping and color jittering are applied.

### 4. Model Architecture



### 5. Training and Evaluation

The model is trained using Adam optimizer with learning rate 1e-4 over 3 epochs. Loss is calculated as the sum of Binary Cross Entropy (real/fake) and Cross Entropy (class prediction).  
  
Evaluation includes calculating two separate accuracy scores: one for real/fake and one for class label prediction.

### 6. Challenges

- Combining spatial and frequency domain data required channel-wise synchronization.  
- FFT computation on GPU increases overhead and memory usage.  
- Handling backbone variability between CNN and ViT while maintaining consistent tensor dimensions.  
- Multi-task learning complexity: optimizing for two tasks with different objectives simultaneously.

### 7. Findings

The results show that integrating FFT with backbone features improves the model's ability to detect deepfakes. The LSTM and attention layers help the model to focus on key spatial regions. The final test set predictions stored in test.csv reflect both label (0/1) and class (object category).  
  
The model achieved high accuracy on both binary and multi-class tasks, demonstrating the effectiveness of FFT + LSTM + attention in a shared backbone architecture.

### 8. Output Artifacts

- Model\_Weights\_Training\_Log.log: Contains saved model weights and training logs.  
- test.csv: File containing predictions in the format [image, label, class].