**Supplementary file**

This file contains details of the transfer learning and deep neural network techniques used for text and image feature extraction with the values of the control parameters.

* 1. **Brief Introduction to techniques used**
* **BERT:** Bidirectional Encoder Representation from Transformers(BERT) is pretrained embeddings model based on transformers. It is bidirectional and combines mask language model with next sentence prediction to better understand the context of the input text.
* **RoBERTa:** Robustly Optimized BERT Pre-training Approach (RoBERTa) is built on BERT’s language masking strategy without next sentence prediction. It uses dynamic masking pattern and is trained over longer sequences making it suitable for NLP tasks.
* **USE:** Universal Sentence Encoder (USE) is sentence encoder to generate embeddings corresponding to the input text. The deep averaging network variant of USE is used. Here the embeddings of input words and their bigrams are fed into a feed forward deep neural network which generates final sentence embeddings.
* **InceptionNet:** InceptionNet is pretrained convolutional neural network with multiscale feature extraction capabilities i.e. it has the ability to extract features from the input data at varying scales by using varying convolutional filters. There constant evolution has led to multiple features of InceptionNet.
* **ResNet:** Residual Network (ResNet) is artificial neural network based model having the capability to skip one or more layers. With the residual blocks the issue training very deep networks is addressed by this model.
* **EfficientNet:** EfficientNet is convolution neural network based model with compound scaling capabilities i.e. all dimensions of depth, width, resolution can be scaled uniformly using a compound coefficient. Eight different variants of this model EfficientNetb0 to EfficientNetb7 have been used in this work. The higher variants work on more number of parameters but this increases training size of the model.
  1. **Parameter values of different techniques and multimodal modal**

Table 1: Parameter values set for different transfer learning and deep neural network techniques

|  |  |
| --- | --- |
| **Technique** | **Parameter values** |
| BERT | Preprocessor: bert\_en\_uncased\_preprocess\_3  Tokenizer: small\_bert\_bert\_en\_uncased\_L-2\_H-256\_A-4\_2/ |
| RoBERTa | Preprocessor: roberta\_en\_cased\_preprocess\_1  Tokenizer: roberta\_en\_cased\_L-12\_H-768\_A-12\_1 |
| USE | Model: universal-sentence-encoder\_4  Tokenizer: PTB Tokenizer |
| InceptionNet | Model: InceptionNetV3 from tensorflow.keras.applications.inception\_v3  Weights: ImageNet |
| ResNet | Model: Resnet50V2  Weights: ImageNet |
| EfficientNet | Version :EfficientNetb0 to Efficientnetb7 from tensorflow.keras.applications |
| Model parameters | Projection\_dims=256  Dropout\_rate=0.1  Num\_projection\_layers=1  Optimiser = Adam  Loss= sparse\_categorical\_crossentropy  Batch\_size=32 |

* 1. **Selection of performance metrics**

MmCAF is evaluated on the following performance criterias:

**Accuracy**: It is the most commonly used metric. Accuracy is determined by the number of correct predictions divided by total predictions. It is a great measure when we have symmetric datasets where values of false positive and false negatives are almost the same.

**F1\_Score**: F1 Score is the weighted average of Precision and Recall. Therefore, this score takes both false positives and false negatives into account. If the cost of false positives and false negatives are very different, it’s better to look at both Precision and Recall.

The time and space complexity of MmCAF is evaluated on the basis of following criteria:

**Number of training parameters:** Number of training parameters denote the number of learnable elements. The parameters can be related to the weights which are learnt during the training of the model.

More number of training parameters can be used to understand complex features of the underlying data.

**Training Time:** Training time is the time required by the model to understand and approximate the relationship between the input data and its label/class. It depends upon multiple aspects like number of parameters, their complexity, batch size, dataset size etc.