5329 Assignment2

QianYu 530458816, Gabriel Goldman 530306632, LinfengYu 530366706

May 17, 2024

Link to Google Drive with our code.

Introduction

What's the aim of the study?

The aim of the study is to train a Deep Learning model for a multi-label classification problem. Being a multi-label and multi-class classification problem poses unique challenges to us meaning we will have to utilise State of the Art (SOTA) Deep Learning techniques to address them. In the push to find a generalised model, much work has been done in the literature to train models on multiple forms of data (Wang et al., 2023). This area of Deep Learning is known as Multi-Modal Learning (MML). The dataset we have been provided contains 40,000 images with 30,000 explicitly labelled and set aside as a training dataset. These images also contain annotations written by humans which allow us to utilise a combination of Computer Vision (CV) and Natural Language Processing (NLP) techniques. This blend of image and text data make this the perfect place to implement a Multi-Modal Learning Model (MMLM).

Why is the study important?

There have been significant strides in Deep Learning, particularly in the the fields of NLP and CV. We see the results of this in our daily lives through our interactions with OpenAI's ChatGPT or facial recognition when unlocking our smartphones. In a bubble these advances are good but not the 0 to 1 technological advances that a generalised model offers. In order to create a generalised Deep Learning model we will need models that are able to handle a variety of different data formats. Using multiple data sources allows MMLMs to enhance the robustness of their predictions and generalise better (Rahate et al., 2022). This study is important because real-world multi-modal applications are prone to noise, missing modalities, or scarcity of multi-modal data (Rahate et al., 2022). MML is an emerging field so having a Kaggle competition to push students to compete and create novel approaches to tackle this problem is of vital importance.

Introduction to our chosen method

Our chosen method utilises a multimodal model where we used Keras' pre-trained Resnet50 model with ImageNet weights. We extracted features from our images using this model and then combined them with the text features which were extracted using a pre-trained ROBERTa model from Hugging Face. We used these combined features as the input for our final model and made predictions on the 19 classes. We used a novel approach to determining the best threshold to predict classes in our data. We verified which threshold gave us the highest validation F1-score as F1-Score is the harmonic mean of precision and recall this should ensure that our model is striking the right balance in it's predictions.

Related Works

Computer Vision

There have been incredible breakthroughs in CV which have been utilised across a number of industries including the food industry (Kakani et al., 2020) where CV has been used to help farmers

by utilising drones to monitor their crops, in radiology to diagnose COVID-19 (Punia et al., 2020) and to automate and streamline many construction management actvities (Paneru and Jeelani, 2021).

A brief history of the breakthroughs is needed. In 2012 Krizhevsky et al. (2012) introduced their ground breaking Convolutional Neural Network (CNN) - AlexNet that won the 2010 LSVRC competition, setting a new benchmark for image classification. Since then further research has utilised different CNN architectures to achieve even more impressive results. Simonyan and Zisserman (2014) introduced VGGNet, which came second in ILSVRC in 2014. Their model showed that utilising deeper networks with smaller filters dramatically increased performance. The winner of 2014's ILSVRC was GoogLeNet by Szegedy (2015) which incorporated "Inception Layers" effectively stacking different sized filters to create variable receptive fields that are able to capture sparse correlation patterns. The winner of ILSVRC 2015 was ResNet by He et al. (2015) which addressed the Vanishing Gradient problem affecting deep neural networks. This is done by using shortcut connections that bypass one or more layers and adds the input of a connection back to the output. There have been further breakthroughs including DenseNet, FractalNet and CapsuleNet (Alom et al., 2021).

In the domain of computer vision, multi-label multi-class image classification represents a significantly more complex problem than traditional single-label tasks. They address complex real-world scenarios where each image may contain multiple objects belonging to different categories. Importantly there are often dependencies between the labels in the image which need to be captured using other techniques (Zhao et al., 2018).

Natural Language Processing

Contemporaneously there have been breakthroughs in NLP across a broad range of tasks such as translation, sentiment analysis and named entity recognition. These breakthroughs have also come from advances in deep learning algorithms, neural network architectures and designs. The first breakthrough in NLP was the Recurrent Neural Network (RNN). RNNs process sequential data by maintaining a hidden state that captures information about the previous words. This allows them to capture long-term dependencies in sequential data. RNNs are defined by the following equations

$$h_t = \sigma(W_{hx}x_t + W_{hh}h_{t-1} + b_h)$$

$$y_t = \phi(W_{hy}h_t + b_y)$$

where:

- x_t is the input at time step t
- h_t is the hidden state at time step t
- y_t is the output at time step t
- W_{hx}, W_{hh}, and W_{hy}
- b_h and b_y are bias vectors
- σ is the activation function
- ϕ is the output activation function

Hochreiter and Schmidhuber's (1997) Long Short-Term Memory (LSTM) model introduced a novel, efficient gradient-based method using gate units that learn to open and close access to constant error flow were a major breakthrough in NLP. Sutskever et al. (2014) introduced their novel encoder-decoder architecture where one LSTM is used to extract features from text representation (encode it) which is then fed to another LSTM network where it is decoded. This model excelled at translation setting a new SOTA benchmark. Vaswani et al. (2014) seminal paper *Attention is all you need* introduced a novel deep learning architecture - the Transformer. This model forgoes recurrence or

convolutional networks and instead utilises a self-attention mechanism. This mechanism works by weighing the importance of different words in a sentence by using a system of queries, keys and values. This allows for parallelisation and long-range dependencies to be captured that were being lost in RNNs.

Using this novel self-attention mechanism, practitioners were able to create Large Language Models (LLMs) such as BERT (Devlin et al., 2018), Meta's Llama, Google's Gemini and OpenAI's ChatGPT. These models use a number of special learning methods including in-context learning and reinforcement learning from human feedback (Chang et al., 2024). These models are trained on billions of parameters using a variety of different methods. The power of these models is their ability to generalise and do well in tasks they were not specifically trained on. This phenomenon is known as 'few-shot' or 'zero-shot' learning where models are not fine-tuned for a specific use case. They are able to perform these tasks based on the context of the training data or semantic relationships (Brown et al. 2020). They have been used in a variety of applications, including medicine (Thirunavukarasu et al., 2023) and storywriting (Yuan et al., 2022). The next step in LLM development is incorporating multiple modes of data.

Multi-Modal Learning

To train and optimise models for more precise and thorough task understanding and prediction, multi-modal learning uses data from multiple modes at once. The model's overall performance and generalisation ability are improved by combining data from several modes, which allows it to capture intricate relationships and features that are challenging to comprehend in a single mode (Huang Y, 2021). The main goal of the multi-modal learning approach is to efficiently integrate the data features from various modes. In their review of multi-modal image classification Jiang et al. (2021) highlight a number of key applications of MML including medical, vision and remote sensing. Much work has been done in this field including OpenAI's newly released ChatGPT4-omni that is able to reason across audio, vision and text in real time.

Data

Our dataset contains 40,000 images and accompanying annotations of which 30,000 are set aside as a training dataset with the labels provided. We further split our training dataset into training and validation set using an 80/20 split.

Exploratory Data Analysis

The dataset contains 19 different labels ranging from 1-19. Figure 1 shows the distribution of the labels across our training dataset. It is important to note that, it is a heavily imbalanced dataset with label 1 occurring 22,797 times. In total there are 46538 labels across our 30,000 images meaning every image has approximately 1.5 labels. Another important thing to note is that there are no occurrences of label 12 in our dataset. We tried removing this value from our model but found that we got worse results on our test set indicating that there are some instances of label 12 in the test set.

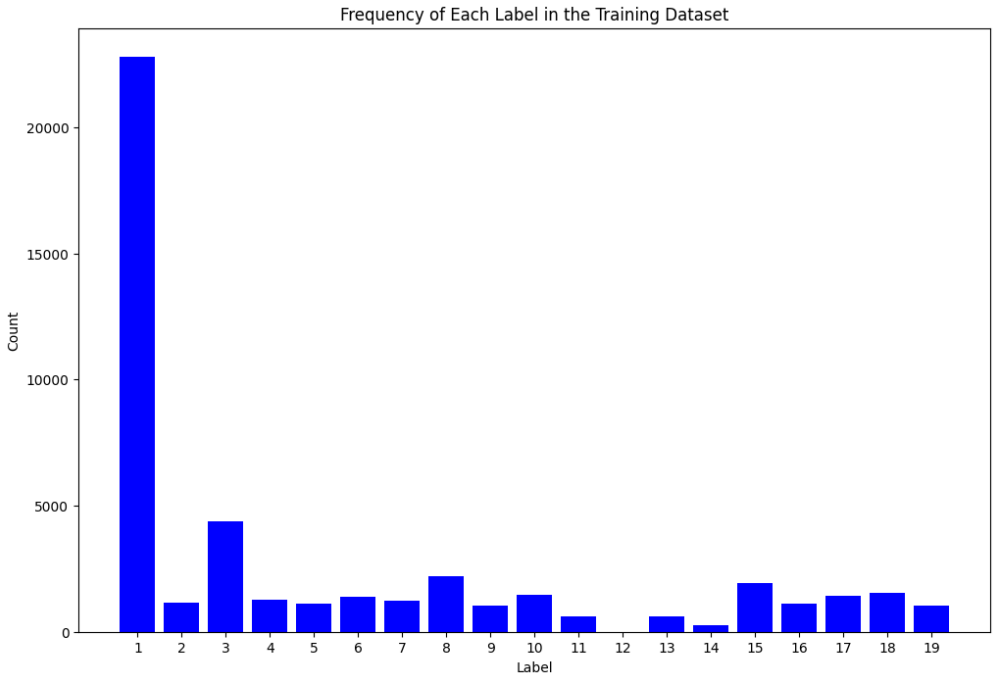

Figure 1: Frequencies of labels in training dataset

As this is a multi-label problem it is important to understand the top combinations and co-occurrences of our labels. Figure 2 shows the top 10 label combinations. We note that label 1 is in all top 10 combinations except the second last in which 3 and 6 occur frequently together.

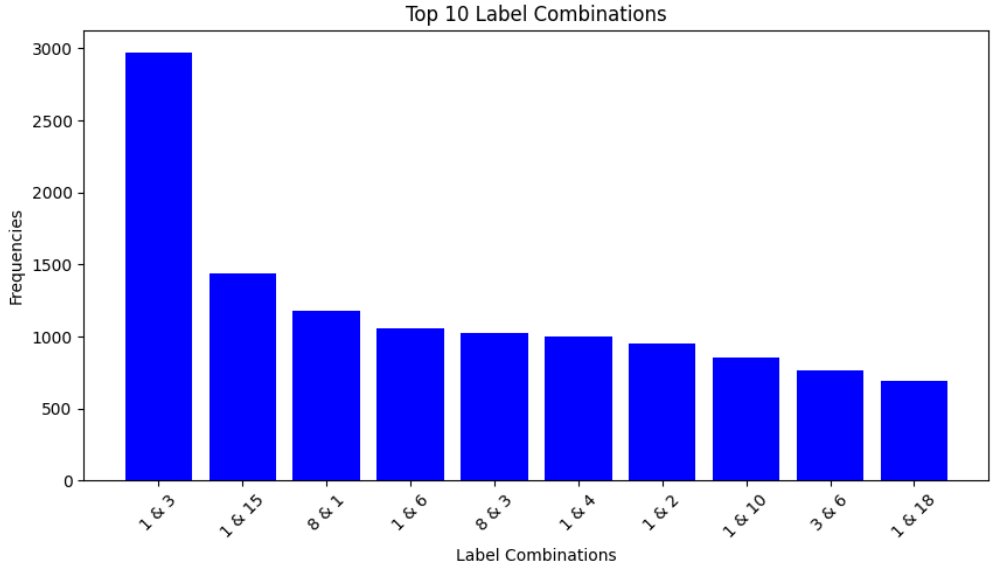

Figure 2: Top 10 Label Combinations

We also looked at the most common words in our annotations after preprocessing them by removing punctuation, removing stopwords, and lemmatising them. Figure 3 shows the top 25 most common words where we can begin to see some kind of labels forming. For example we see "man", "woman", "people" and "person" in the top 25. However we will need to utilise deep learning methods to truly classify our images.

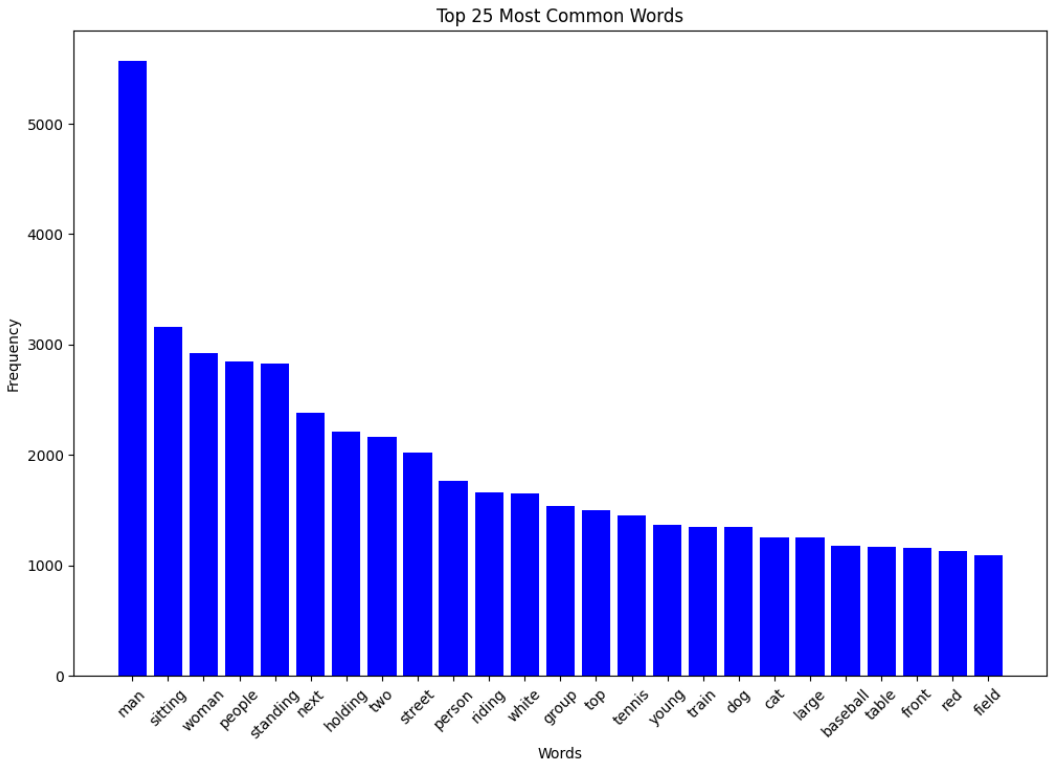

Figure 3: Top 25 Word Combinations

Pre processing

The images in our dataset are of an inconsistent size and require preprocessing prior to training in our model. We decided to also augment our images as this has been shown to help a model learn the features of an image. We used functions from the Tensorflow library to do this. Particularly we randomly cropped, flipped, rotated and adjusted the brightness and saturation.

The text data was preprocessed using the Roberta Tokenizer as, this pre-trained model requires particular preprocesssing. We one-hot encoded our labels, meaning each label had an array of length 19 where 1 on that particular index indicates the image has one of these labels. This is common practice in multi-label classification.

Techniques

Principles behind chosen technique

We leverage both image and text data to enhance the classification performance in our multi-label classification task. Our model is able to extract important information from images by combining features with the textual descriptions that go along with them.

Image feature extraction: For image feature extraction, we use the ResNet50 model. ResNet50 is pre-trained on the ImageNet dataset, allowing us to use the representations it learns to accomplish our specific tasks. Below are the steps we mainly use for image feature extraction.

1. Preprocessing: The images are first preprocessed to ensure consistency in input size and normalization. This includes resizing the images to 224×224 pixels, normalizing pixel values, and applying data augmentation techniques such as random flipping, brightness adjustment, cropping,

rotation, and saturation changes to enhance the model's robustness.

$$\text{image} = \frac{\text{image} - \text{mean}}{\text{std}}$$

2. Feature Extraction: The preprocessed images are passed through the ResNet50 model to extract deep features. These features are then pooled using global average pooling to obtain a compact representation of each image.

$$\text{features}_{\text{image}} = \text{GlobalAveragePooling}(\text{ResNet50}(\text{preprocessed\_image}))$$

Text Feature Extraction: For text feature extraction, we adopt the RoBERTa model. The specific introduction will be mentioned in the Techniques section. RoBERTa is an upgrade on the BERT model that was trained on a much larger dataset, for a longer time and using more dynamic learning techniques. The tokenized text is input into the RoBERTa model, and the last hidden state of the [CLS] token is extracted as a feature vector representing the entire input text.

$$\text{features}_{\text{text}} = \text{RoBERTa}_{\text{CLS}}(\text{tokenized\_text})$$

Integration of Image and Text Features: The extracted image and text features are integrated into a multi-modal model. This integration enables the model to leverage both visual and textual information for improved classification performance (Xuri Ge, 2021).

1. Feature Processing: Both image and text features are processed through fully connected layers with batch normalization and dropout to enhance generalization and reduce overfitting.

$$x_{\text{img}} = \text{Dropout}(\text{BatchNormalization}(\text{Dense}(512, \text{ReLU})(\text{features}_{\text{image}})))$$

$$x_{\text{txt}} = \text{Dropout}(\text{BatchNormalization}(\text{Dense}(512, \text{ReLU})(\text{features}_{\text{text}})))$$

2. Feature Fusion: The processed image and text features are then concatenated and further processed through additional fully connected layers. We apply multi-head attention to capture interactions between the combined features.

$$\text{concatenated} = \text{Concatenate}([x_{\text{img}}, x_{\text{txt}}])$$

$$\text{transformer\_output} = \text{MultiHeadAttention}(\text{num\_heads} = 2, \text{key\_dim} = 512)(\text{concatenated}, \text{concatenated})$$

$$x = \text{Dense}(256, \text{ReLU})(\text{transformer\_output})$$

$$x = \text{BatchNormalization}(x)$$

$$x = \text{Dropout}(0.2)(x)$$

$$x = \text{Dense}(128, \text{ReLU})(x)$$

$$x = \text{BatchNormalization}(x)$$

$$x = \text{Dropout}(0.2)(x)$$

$$\text{outputs} = \text{Dense}(19, \text{sigmoid})(x)$$

3. Model Compilation and Training: Finally, the Adam optimizer and binary cross entropy loss are used to compile the model. The Adam optimizer and binary cross entropy loss are key components in training our multi-modal model, where the robustness and efficiency of the Adam optimizer are better suited to our multi-label classification task. At the same time, the use of binary cross entropy loss can independently process each label and effectively punish each label for false predictions. By encouraging the model to produce a probability output that matches the true distribution (L Feng, 2021), these two approaches maximize the advantages of the model.

$$\text{loss} = -\frac{1}{N}\sum_{i=1}^{N} y_i \log(\hat{y}_i) + (1 - y_i)\log(1 - \hat{y}_i)$$

This is the basic framework of our multi-modal learning, which can effectively use visual and text information to improve classification performance in multi-label classification tasks. This approach

ensures that the model can capture complementary information from both data modes, enhancing its ability to understand and classify complex data sets. Figure 4 shows the architecture of our chosen model.

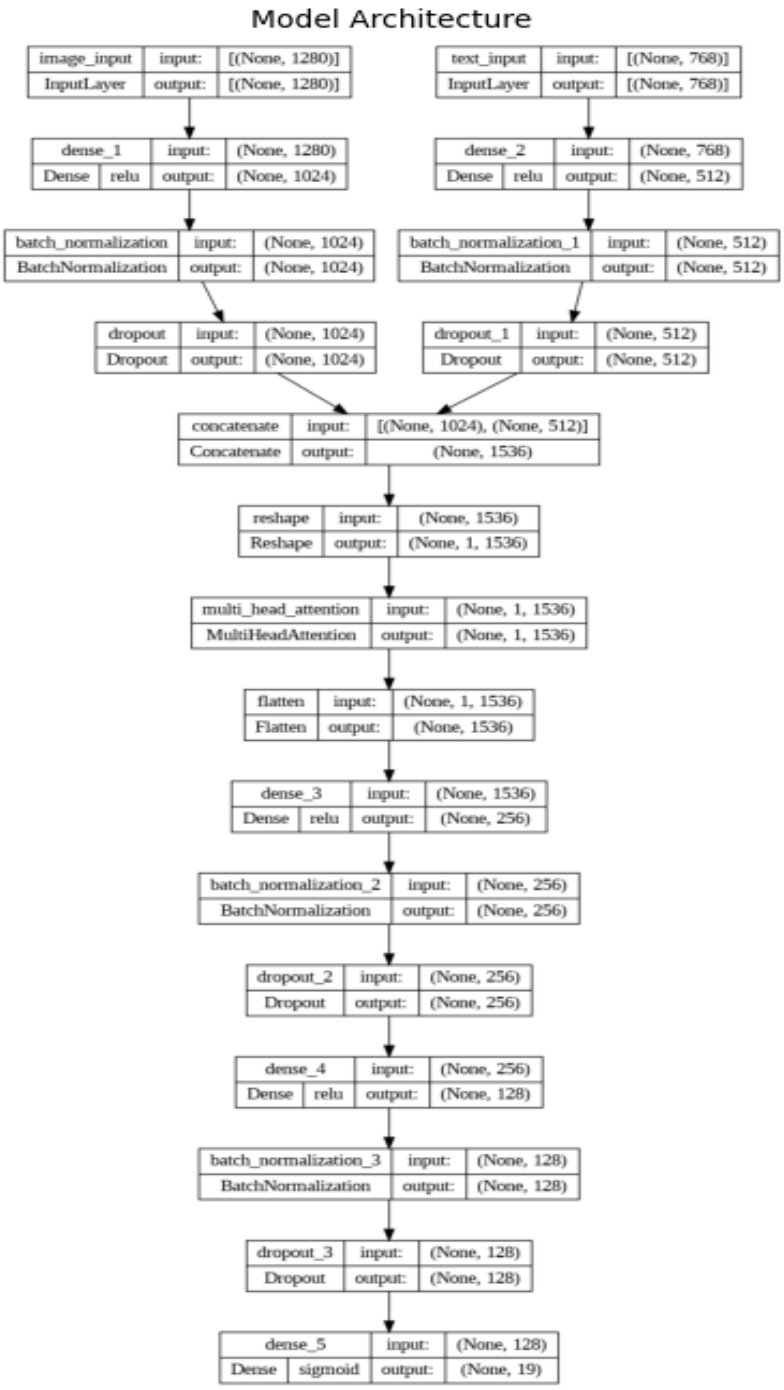

Figure 4: Accuracy

Justification for chosen technique

1. Text Feature Extraction: Natural Language Processing (NLP) techniques are applied to the text data associated with each image in this multi-label classification task. Because RoBERTa is well known for its capacity to identify contextual relationships in text, it is an excellent option for obtaining significant information from picture captions. We use the Hugging Face Transformers library's pre-trained roberta-base-uncased model. The RoBERTa model offers a profound bidirectional comprehension of language, which is essential for accurately identifying the subtleties and context found in textual data.

The text data is tokenized using the RobertaTokenizer, which also handles padding and truncation to guarantee consistent input sequence lengths. This process turns the text into tokens that the RoBERTa model can comprehend. After tokenizing the text, features are extracted using the RoBERTa model. To be more precise, the last hidden state of the [CLS] token—which functions as a thorough representation of the whole input text—is obtained using the TFRoertaModel. Text is processed by the RoBERTa model by first encoding it into tokens, which are then passed through a number of transformer layers, each of which consists of feed-forward neural networks and a multi-head self-attention mechanism. By enabling the model to dynamically focus on various segments of the input text, the attention mechanism improves the model's comprehension of context.

The text feature vector is derived from the RoBERTa model's final output, namely the representation of the [CLS] token. Next, we incorporate the extracted text features into our multi-modal model. These features are further processed by a fully connected layer prior to being concatenated with the image features. The model can now better perform in classification by utilising both textual and visual information thanks to this integration. We guarantee that our model extracts rich contextual information from the captions by using RoBERTa for text processing, which greatly improves the overall performance of our multi-label classification task.

2. Image Feature Extraction: We utilise a pre-trained Resnet50 model to extract the features from our image. Resnet50 is a SOTA deep learning model that excels at image classification making it a perfect candidate for extracting features from our dataset. This model is downloaded using the Tensorflow API. We use a pre-trained model, trained on the ImageNet dataset. We freeze the layers and do not include a top so that the model will only output the features of our images in vector arrays.

Bidirectional Contextual Understanding:

- **Transformer Architecture:** RoBERTa uses a Transformer architecture with multi head self attention mechanisms, allowing it to capture the context of a word based on both its left and right surroundings. This bidirectional understanding is crucial for interpreting the meaning and nuances of text, especially in complex label text(Koroteev, 2021).
- **Contextual Embeddings:** Contextual embeddings are produced by RoBERTa, in which each word's representation is contingent upon the entirety of the sentence. This enables the model to effectively handle other complex language phenomena, such as the words with multiple meanings.

Advantages of Integrating Image and Text Features

- **Complementary Information:**The model can leverage complementary information from different modalities, resulting in more robust and accurate predictions, by combining features from both images and text.
- **Contextual Richness:** The textual features extracted by BERT provide rich contextual information that complements the visual features extracted by ResNet50, enabling the model to better understand and classify complex data.

Advantages or novelty of chosen technique

1. Advantages: We utilise a unique MMLM that utilises data from both images and text. We utilise pre-trained models extensively to extract the features from our data set. Many compute hours and millions of dollars have been spent training these models. Our model uses Multi-head attention to learn the combined image features and text features.

We are compute constrained having to rely on either availability of Google Colab's free GPU or sparingly use their GPU A100. This allows us to focus most of our time on training our final model architecture saving us both compute and time.

2. Novelty: Our model is novel, particularly in it's use of tuning the threshold for classification. For a multi-label classification problem it is not as trivial as a single-label classification problem where an image can only be label or another. We used validation f1-score from training the model and optimised our threshold for this. Additionally, we used a Multi-head attention layer to interpret the combined outputs from our pre-trained model. This promotes better generalisation as the model is able to focus on multiple parts of the input data simultaneously. Each part can learn a different feature allowing our model to better understand the relationship between the features.

3. Disadvantages: The major disadvantage to our approach is that it highly computationally expensive to extract the image features from all of our images and text data. We mitigated this by saving the features but this hindered our experimentation in terms of preprocessing. Another disadvantage to using pretrained models is that we cannot fine-tune the model during training. We could have tried to train some layers of the model but this would have massively increased the computation cost. Another peculiarity is that the validation f1-score we achieved was lower than what we were achieving on our test set. This may be due to the distribution of the test set or due to the imbalance of the training and validation sets.

Accuracy and Efficiency

Efficiency was vitally important in all of our models as we were constrained by our compute. Our best model ran trained for 4 seconds per epoch and only ran for 46 epochs (approximately 3 minutes) before hitting our early stopping parameter. This is even after using our decaying learning rate scheduler. This model was trained using Google Colab's A100 GPU which is optimised for deep learning.

Accuracy Figure 5 shows the training and validation F1 scores of our best model. The training F1 score rose rapidly during the initial epochs and eventually stabilized around 0.85. In contrast, the validation F1 score fluctuated significantly in the early stages but generally increased, ultimately stabilizing around 0.8. Both trends show stability, with consistent performance on both the training and validation sets. The lack of significant overfitting indicates that the model performs well.

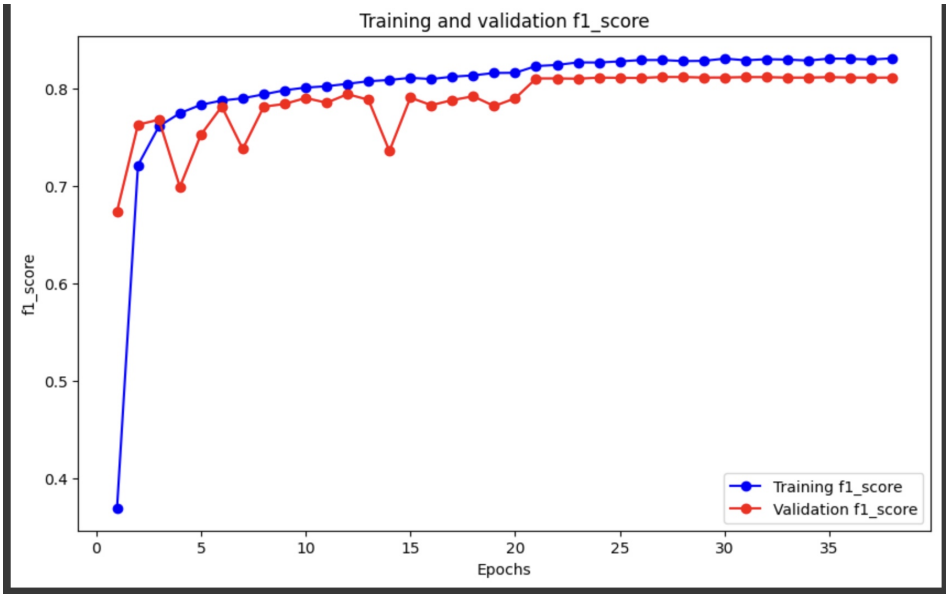

Figure 5: Adam F1-Score of best model

Extensive Analysis: Ablation Studies

We underwent comprehensive ablation studies when creating our best model. We made 23 submissions to Kaggle, gradually improving our performance. In particular we examined the effect of dropout rate, finding that a large dropout rate lead to much worse performance as shown when comparing Figure 6 - where Dropout is zero and Figure 7 - where dropout is 0.5. We see that they achieve similar validation accuracy's but the validation loss when we have 0.5 does not go as low as when we have no dropout. Similarly when we have 0.5 dropout the training accuracy is not very good with clear signs of underfitting. We decided to do some more extensive studies on our dropout rate trying a range of different values which we will talk about in our hyperparameter analysis.

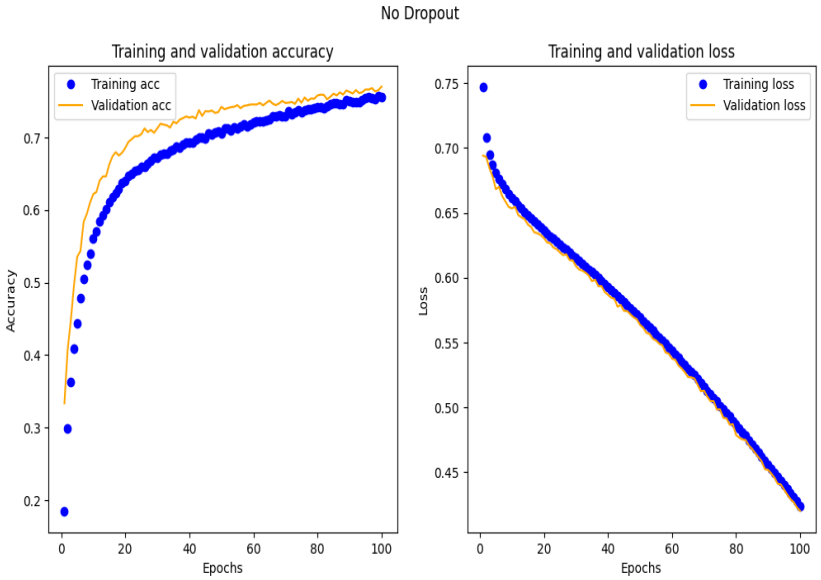

Figure 6: Model with dropout rate == 0.0

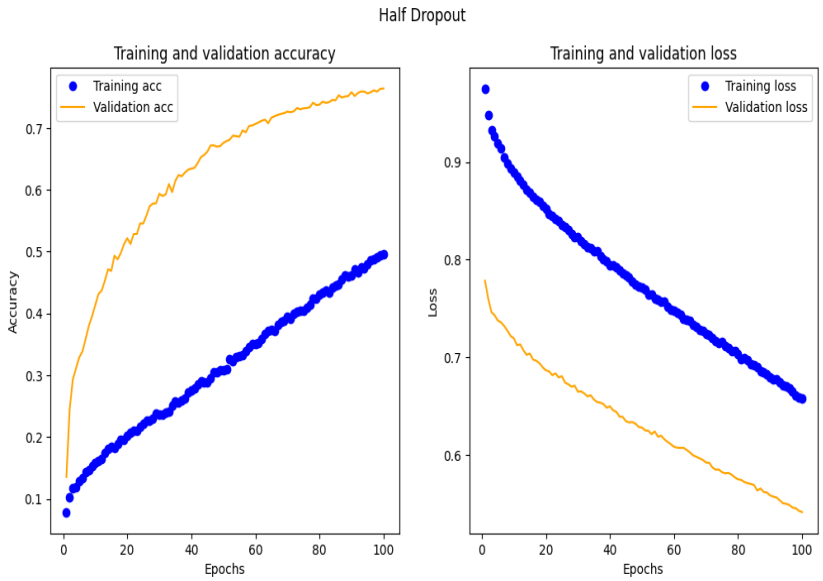

Figure 7: Model with dropout rate == 0.5

We tried modifying our loss function to account for the class imbalance. We implemented a custom weighted loss function which, finds the proportion of each label in our dataset. The loss function is then weighted extra for these minority classes making the model pay more attention to them. Although this seemed like a logical idea, in practice we found that this did not have a big impact on model performance. We also tried using the class weights function of Keras' fit function but found that this had a negative impact on the performance of our model.

We tried removing the empty label from our one hot encoded labels as we did not want our model to predict it as there were no instances of it in the training data. When we did this and uploaded to Kaggle we found that we were actually getting worse results. We then changed our model back to having 19 classes instead of 18.

We tried removing the BatchNormalisation layers from our model to see the impact that this would have on our model. Batch normalisation can accelerate training and regularise our model but as our model is not very deep we wanted to see the impact this would have. We found that this led to worse results confirming, inline with the literature the importance of Batch normalisation.

We tried removing the empty label from our one hot encoded labels as we did not want our model to predict it as there were no instances of it in the training data. When we did this and uploaded to Kaggle we found that we were actually getting worse results. We then changed our model back to having 19 classes instead of 18.

We tried using different pre-trained models for our feature extraction. We tried using ResNet101 and ResNet152 but found that there was very minor or no impact on performance and feature extraction used considerably more compute so we decided to change back to the smaller ResNet50. For our text feature extraction we initially used BERT which did a good job earning a F1-Score on Kaggle of 89.61. This was a good result but we decided to try and use RoBERTa as we had heard it outperformed BERT. This made a huge impact bumping our F1-Score to 92.186.

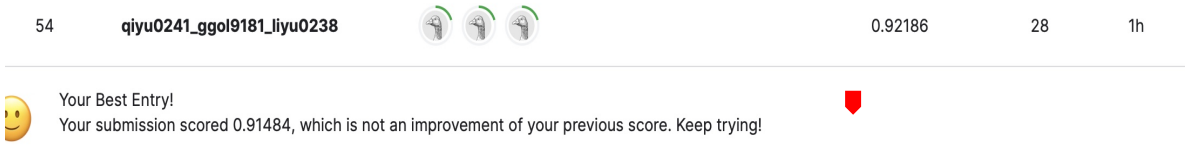

Figure 8: The best Kaggle score

Extensive Analysis: Comparison Methods

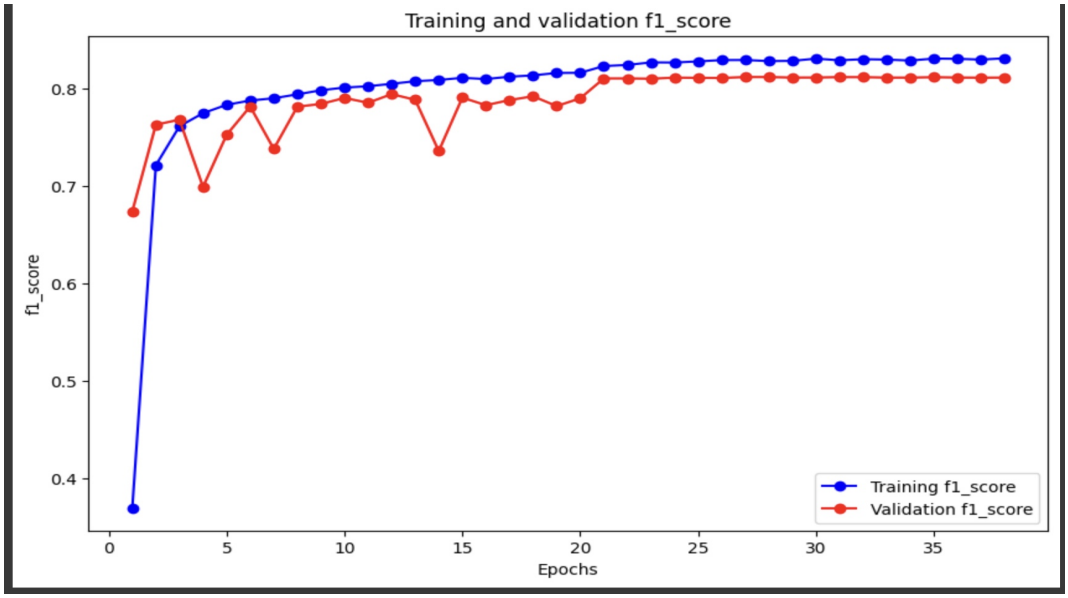

Figure 9: Adam Accuracy

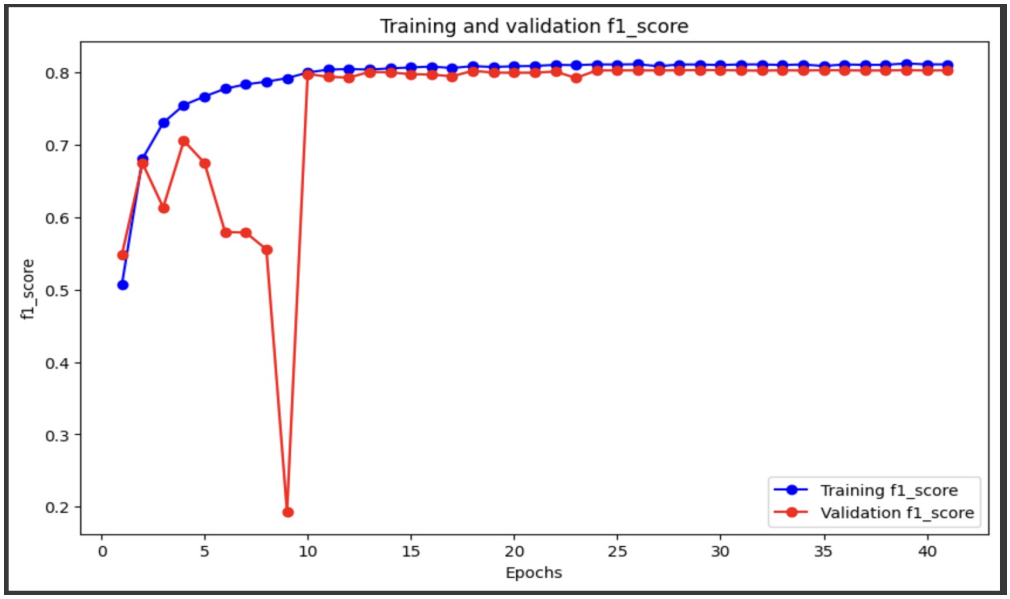

Figure 10: SGD Accuracy

The two plots above demonstrate the training and validation accuracy of our multi-modal model over multiple epochs using two different optimization algorithms: Adam and SGD .

Adam

- **Stable Performance:** After the initial rapid increase, the training accuracy stabilizes around 0.82 to 0.85. The validation accuracy also stabilizes around 0.80, with some fluctuations. This close alignment between training and validation accuracy suggests minimal overfitting and good generalization.

SGD

- **instable Learning Curve:** The learning curves for the SGD optimizer show more variability and instability, especially in the initial epochs, indicating a less consistent learning process.

Overall, the Adam optimizer provides a more robust and efficient training process with better stability, faster convergence, and higher generalization ability than the SGD optimizer. These features make Adam the first choice for training multimodal models in this multi-label classification task.

Extensive Analysis: Hyperparameter Analysis

As we were constrained by our compute resources we did not undergo a comprehensive grid search on all the parameters in our model. We used a decaying learning rate scheduler and early stopping with model checkpoints to regulate our model during training and ensure it does not overfit. The actual architecture of our model is not very complex meaning there are not actually that many hyper-parameters to tune.

We did undergo extensive tuning of our dropout value, trying values ranging from 0.0 to 0.5 and found a value 0.1 led to the highest validation f1-score. We tried a number of different starting learning rates ranging from 0.0001 to 0.0005 decreasing by a factor of 0.0001. We found that the lowest learning rate of 0.0002 led to the best results. We tried different values in our learning rate scheduler by adjusting the factor and minimum learning rate values in Tensorflow's ReduceLROnPlateau function. We tried a very low minimum learning rate (10^{-7} and high minimum rate learning rate (10^{-3} but found that setting the minimum learning rate to be 10^{-6} led to the best performance and fastest convergence. We also tried a range of values for the factor hyperparameter and found that 0.05 was the best.

Discussion and Conclusion

The multimodal learning method combined with RoBERTa and ResNet50 proved to be effective for multi-label classification tasks. The integration of text and image features enables the model to capture complementary information from both patterns, improving performance. The use of the Adam optimizer further enhances the training process.

Our approach demonstrates the potential of multi-modal learning to solve complex classification tasks, leveraging the strengths of textual and visual data to achieve robust and accurate predictions. We add to the literature by showcasing the power of utilising pre-trained models for feature extraction. This is a burgeoning field in deep learning (Liang et al., 2022) and we feel our novel threshold approach is a great contribution. Additionally using Multi-head attention to interpret the combined text and image data is a novel approach which shows great results. Future work could explore further enhancements, such as fine-tuning the choice of hyperparameter settings for RoBERTa and ResNet50 models, and experimenting with other advanced optimization algorithms to further improve performance.

References

Wang, X., Chen, G., Qian, G., Gao, P., Wei, X.-Y., Wang, Y., Tian, Y., & Gao, W. (2023). Large-scale Multi-modal Pre-trained Models: A Comprehensive Survey. Machine Intelligence Research, 20(4). https://doi.org/10.1007/s11633-022-1410-8

Jain, D. K., Rahate, A., Joshi, G., Walambe, R., & Kotecha, K. (2022). Employing Co-Learning to Evaluate the Explainability of Multimodal Sentiment Analysis. IEEE Transactions on Computational Social Systems, 1–8. https://doi.org/10.1109/tcss.2022.3176403

Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2012). ImageNet Classification with Deep Convolutional Neural Networks. Communications of the ACM, 60(6), 84–90. https://doi.org/10.1145/3065386

Ravneet Punia, Kumar, L., Mujahid, M., & Rajesh Rohilla. (2020). Computer Vision and Radiology for COVID-19 Detection. 2020 International Conference for Emerging Technology (INCET). https://doi.org/10.1109/incet49848.2020.9154088

Zhao, B., Li, X., Lu, X., & Wang, Z. (2018). A CNN–RNN architecture for multi-label weather recognition. Neurocomputing, 322, 47–57. https://doi.org/10.1016/j.neucom.2018.09.048

Simonyan, K., & Zisserman, A. (2015). Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:1409.1556. Retrieved from https://arxiv.org/abs/1409.1556

Kakani, V., Nguyen, V. H., Kumar, B. P., Kim, H., & Pasupuleti, V. R. (2020). A critical review on computer vision and artificial intelligence in food industry. Journal of Agriculture and Food Research, 2(1), 100033. https://doi.org/10.1016/j.jafr.2020.100033

Paneru, S., & Jeelani, I. (2021). Computer vision applications in construction: Current state, opportunities & challenges. Automation in Construction, 132, 103940. https://doi.org/10.1016/j.autcon.2021.103940

Hochreiter, S., & Schmidhuber, J. (1997). Long Short-Term Memory. Neural Computation, 9(8), 1735–1780. https://doi.org/10.1162/neco.1997.9.8.1735

Alom, M. Z., Taha, T. M., Yakopcic, C., Westberg, S., Sidike, P., Nasrin, M. S., Van Essen, B. C., Awwal, A. A. S., & Asari, V. K. (2018). The history began from AlexNet: A comprehensive survey on deep learning approaches. arXiv preprint arXiv:1803.01164. Retrieved from http://arxiv.org/abs/1803.01164

Sutskever, I., Vinyals, O., & Le, Q. V. (2014). Sequence to sequence learning with neural networks. arXiv preprint arXiv:1409.3215. Retrieved from http://arxiv.org/abs/1409.3215

Devlin, J., Chang, M.-W., Lee, K., & Toutanova, K. (2018). BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. CoRR, abs/1810.04805. Retrieved from http://arxiv.org/abs/1810.04805

Huang, Y., Du, C., Xue, Z., Chen, X., Zhao, H., & Huang, L. (2021). What makes multi-modal learning better than single (provably). Advances in Neural Information Processing Systems, 34, 10944-10956.

Chang, Y., Wang, X., Wang, J., Wu, Y., Yang, L., Zhu, K., Chen, H., Yi, X., Wang, C., Wang, Y., Ye, W., Zhang, Y., Chang, Y., Yu, P. S., Yang, Q., & Xie, X. (2024). A survey on evaluation of large language models. ACM Transactions on Intelligent Systems and Technology, 15(3), Article 39, 45 pages.

Brown, T. B., Mann, B., Ryder, N., Subbiah, M., Kaplan, J., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., Agarwal, S., Herbert-Voss, A., Krueger, G., Henighan, T., Child, R., Ramesh, A., Ziegler, D. M., Wu, J., Winter, C., ... Amodei, D. (2020). Language models are few-shot learners. arXiv preprint arXiv:2005.14165. Retrieved from https://arxiv.org/abs/2005.14165

Thirunavukarasu, A. J., Ting, D. S. J., Elangovan, K., & others. (2023). Large language models in medicine. Nature Medicine, 29(1930-1940). https://doi.org/10.1038/s41591-023-02448-8

Xuri Ge, Fuhai Chen, Joemon M. Jose, Zhilong Ji, Zhongqin Wu, and Xiao Liu. 2021. Structured Multi-modal Feature Embedding and Alignment for Image-Sentence Retrieval. In Proceedings of the 29th ACM International Conference on Multimedia (MM '21). Association for Computing Machinery, New York, NY, USA, 5185–5193. https://doi.org/10.1145/3474085.3475634

Yuan, A., Coenen, A., Reif, E., & Ippolito, D. (2022). Wordcraft: Story writing with large language models. In Proceedings of the 27th International Conference on Intelligent User Interfaces (pp. 841–852). Association for Computing Machinery. https://doi.org/10.1145/3490099.3511105

Feng, L., Shu, S., Lin, Z., Lv, F., Li, L., & An, B. (2021). Can cross entropy loss be robust to label noise? In Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence (IJCAI). https://www.ijcai.org/proceedings/2020/0305.pdf

Yeung, M., Sala, E., Schönlieb, C. B., & Rundo, L. (2022). Unified focal loss: Generalising dice and cross entropy-based losses to handle class imbalanced medical image segmentation. Computerized Medical Imaging and Graphics, Elsevier. https://www.sciencedirect.com/science/article/pii/S0895611121001750

Jiang, X., Ma, J., Xiao, G., Shao, Z., & Guo, X. (2021). A review of multimodal image matching: Methods and applications. Information Fusion, 73, 22–71. https://doi.org/10.1016/j.inffus.2021.02.012

Koroteev, M. V. (2021). BERT: a review of applications in natural language processing and understanding. arXiv preprint arXiv:2103.11943.

Liang, T., Lin, G., Wan, M., Li, T., Ma, G., & Fengmao Lv. (2022). Expanding Large Pre-trained Unimodal Models with Multimodal Information Injection for Image-Text Multimodal Classification. 2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). https://doi.org/10.1109/cvpr52688.2022.01505

Appendix

AI tools such as OpenAI's ChatGPT 3.5,4 and 4o and Google Colab's inbuilt AI were used throughout the writing of this report to help debug code and explain concepts.