Mini Project Report

on

Multi-Modal method for person behavior identification using social media images

Submitted by

Ketan Bhadwariya **20bds032**

Omkar Gowda **20bds022**

Vipul Bawankar **20bds063**

Navneet Sen **20bds037**

Under the guidance of

Dr. Pavan Kumar C

Head of Department, Computer Science Engineering

DEPARTMENT OF DATA SCIENCE AND INTELLIGENT SYSTEMS

INDIAN INSTITUTE OF INFORMATION TECHNOLOGY DHARWAD

08/05/2023

Contents

List of Figures

List of Tables

1 Introduction

The use of social media platforms has become an integral part of people's daily lives, and this has led to the generation of large volumes of multimedia data. With the increasing popularity of social media platforms, there is a growing need to develop effective methods to analyze the vast amount of data that is being generated. One area where this is particularly important is in identifying person behavior patterns from social media images.

In recent years, there has been a growing interest in the development of multi-modal methods for person behavior identification using social media images. These methods use a combination of different data sources, such as image features, and text metadata, to identify patterns in person behavior. By leveraging multiple data sources, these methods can provide more accurate and comprehensive insights into person behavior than traditional single-modal approaches.

In this work, we have integrated visual and textual data by extracting image and text embeddings separately, concatenating them, and passing them into a neural network. We have used VGG16 and BERT for image and text embeddings respectively. We have also explored the effectiveness of passing the embeddings separately into neural networks and then concatenating the intermediate output. Our aim is to classify images into different categories based on their descriptions. We have used a dataset consisting of images and corresponding textual descriptions from Flickr30k [5]. The results show that the concatenated approach outperforms the separate approach, achieving an accuracy of 80%. This work demonstrates the effectiveness of integrating visual and textual data using deep learning techniques and highlights the potential for further exploration in this area.

2 Related Work

Since we can hardly find techniques focused on the classification of behavior-oriented social media images as mentioned above, we review the methods, Tiwari et al. [7] proposed suspicious face detection based on eye and other facial features movement monitoring. The method explores facial features to classify persons as normal or abnormal. When a person is abnormal, facial features such as eye, torso, gesture, and speech behave in an abnormal way. The method uses non-linear entropy to extract such abnormal behaviors for classification. Barsoum et al. [1] proposed a deep learning model for facial expression recognition based on crowd-sourced label distribution. The method uses multi-label information for discarding noises and false labels such that the performance of the method improves significantly. However, the method accepts cropped face images of different expressions for achieving results. Sharma et al. [6] proposed a method for emotion recognition using keypoint descriptors and texture features. Mungra et al. [4] proposed a CNN-based method for emotion recognition using facial expressions. The method performs several pre-processing steps to handle variations on facial expression images. Then CNN has been trained for the classification of emotions. The performance of the method depends on the success of pre-processing steps. There are high chances of losing vital information of facial expressions because of pre-processing steps. In summary, the above methods on the classification of emotions work well when the input images have cropped faces of different expressions, but not the images where multiple expressions are present in a single image. In addition, the scope of the methods is limited to emotion recognition and classification but not personality traits classification.

To overcome this problem, Liu et al. [3] proposed a method for personality traits classification from the full images but not cropped face images. The method combines Twitter content, text, and image features for achieving results. The method is limited to five classes of personality traits. Since the content of Twitter is not available in the case of our work, the method may not be robust and effective for the classification of ten classes. Krishnani et al. [2] proposed a structural function-based transform feature for the classification of behavior-oriented social media images. The method detects a face in the input image and uses facial key points for feature extraction.

3 Data and Methods

3.1 Dataset Preparation

The problem we faced in developing a multi-modal method for person behavior identification using social media images was the lack of availablity of dataset that contained both text and images with labels for this specific problem. We needed a dataset that could be used for training and testing our method.

To address this issue, the we utilized the Flickr30k Dataset, which is a well-known dataset for sentence-based image description. This dataset contains a large number of images with corresponding textual descriptions and appropriate image captioning. We obtained this dataset from Kaggle, which is a platform for data scientists to find and share datasets and code.

To make the Flickr30k Dataset usable for our specific problem, we manually labeled 100 data points into 6 different labels that were relevant to person behavior identification. The labels were based on different types of behaviors that people might exhibit in social media images. This included labels: Images of Animals, Kids, Group Photos, Solo pictures, People playing Sports, and People Working. After labeling the data, we stored it in a .csv file and created a new dataset with four features or columns: ID, Image Path, Image Description, and Label. The ID column contained a unique identifier for each data point, the Image Path column contained the path to the corresponding image file, the Image Description column contained the textual description of the image, and the Label column contained the behavior label assigned to that particular data point.

By using the Flickr30k Dataset and manually labeling data, we were able to create a dataset that met their needs and allowed us to train and test their multi-modal method for person behavior identification using social media images.

3.2 Methodology

We initially worked on text and image data separately, treating them as single modalities. Later, we utilized fusion techniques to combine the features extracted from both modalities and perform multi-modal classification. This approach gave accurate and comprehensive analysis of person behavior patterns in social media images.

3.2.1 Text dataset

(a) **Preprocessing:**

i. **Removing special characters:** Special characters such as punctuation marks, brackets, and mathematical symbols were removed from the text. This helps to eliminate noise from the data and improve its readability.

ii. **Removing ".jpg" from image description:** We removed the file extension ".jpg" from the image descriptions. This was necessary because the presence of the file extension could confuse the model and affect its performance.

iii. **Uppercase to lowercase:** The text was converted to lowercase to ensure consistency and reduce the number of unique words in the dataset.

iv. **Removing stopwords:** Stopwords are common words such as "the," "a," and "an" that do not add much meaning to the text. Removing them helps to reduce the size of the vocabulary and improve the efficiency of the model.

v. **Lemmatization:** This involves reducing words to their base form (lemmas). For example, the word "running" would be reduced to "run." This helps to reduce the number of unique words in the dataset and improve the accuracy of the model.

vi. **Visualisation:** We visualized the textual data using a word cloud and a histogram of word frequencies. The word cloud displays the most frequently occurring words in the dataset, with the size of each word representing its frequency. The histogram of word frequencies displays the number of occurrences of each word in the dataset.

(b) **BERT Embeddings:**

Bidirectional Encoder Representations from Transformers (BERT) is a powerful pre-trained language model developed by Google researchers in 2018. In our study, we used the bert-base-uncased version of the model, which has 110 million parameters. The model was pre-trained on a massive amount of text data and is capable of encoding text into high-dimensional vector representations, called embeddings.

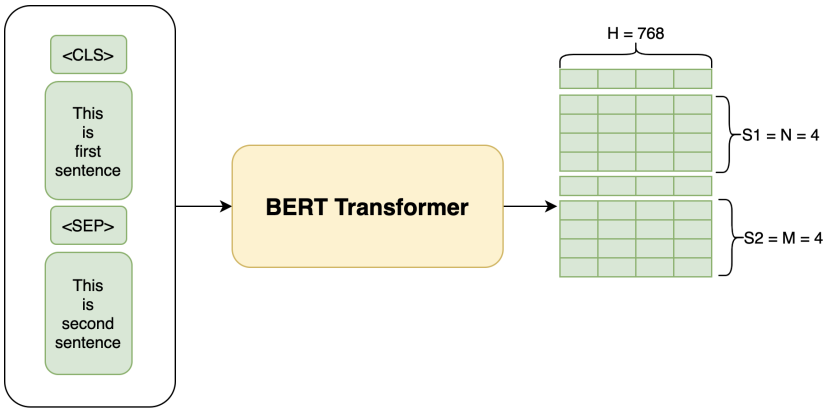

Figure 1. BERT embeddings CLS, MASK, SEP, PAD

To use BERT for our specific task of person behavior identification using social media images, the textual data was tokenized using the AutoTokenizer from the transformers library. Then we loaded the pre-trained BERT model and added special tokens to the text, including [CLS] (classification), [MASK] (masking), [SEP] (separating sentences), and [PAD] (padding). These tokens help the model understand the structure and context of the text. The BERT model was then used to extract embeddings from the text data, with each embedding having a size of 768. After extracting the BERT embeddings for the text data, we used t-SNE (t-Distributed Stochastic Neighbor Embedding) to visualize the embeddings in a 2-dimensional and 3-dimensional vector space.

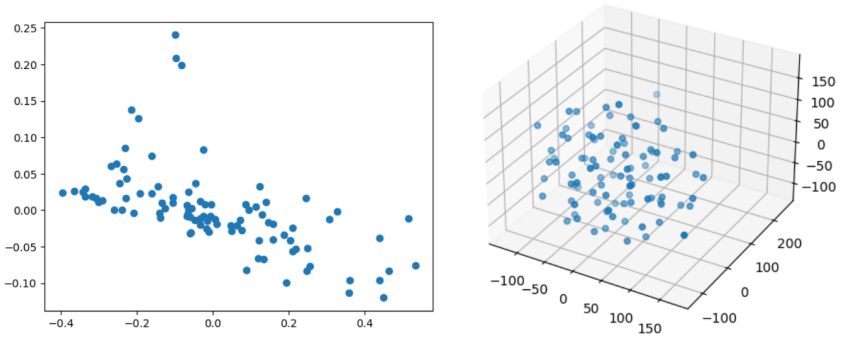

Figure 2. t-SNE to visualize BERT embeddings in 2-dimensional and 3-dimensional vector space

(c) **Model:** After preprocessing the data and extracting the BERT embeddings, we split the data into training and testing sets using the train_test_split function from the sklearn library and then label encoded the behavior labels and used logistic regression to classify the behavior based on the text and image data.

The model achieved an accuracy of 65% on the test data, meaning that it correctly classified 65% of the data points in the test set. This result was a decent starting point for us and can be improved by using image data also.

3.2.2 Image dataset

(a) **Preprocessing:** In the preprocessing step, we loaded the images from the given paths and visualized them. We resized all images to a fixed size of (250, 250, 3), where 250 and 250 are the height and width of the image, respectively, and 3 corresponds to the RGB color channels. The resizing is necessary because the images in the dataset can have different sizes and dimensions, and machine learning models require input data of a consistent size and shape.

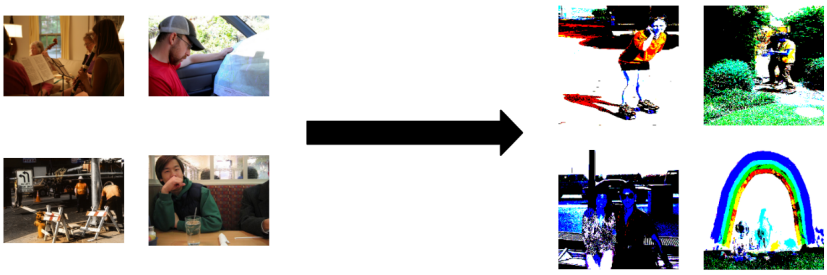

Figure 3. Images before and after resizing

Further we converted the images from .jpg format to arrays, and added an extra dimension to each image for VGG processing. The additional dimension corresponds to the batch size, which is a requirement of the VGG16 model used in this study. Finally, we used the preprocess_input function from the TensorFlow library to preprocess the image data for VGG16, which involves subtracting the mean RGB value of the training set from each pixel of the image.

(b) **VGG16 embeddings:**

VGG16 embeddings are a type of image feature extraction technique obtained from the VGG16 convolutional neural network model that has been pre-trained on a large dataset called ImageNet. The weights obtained after training on ImageNet data are used as the starting point for VGG16.

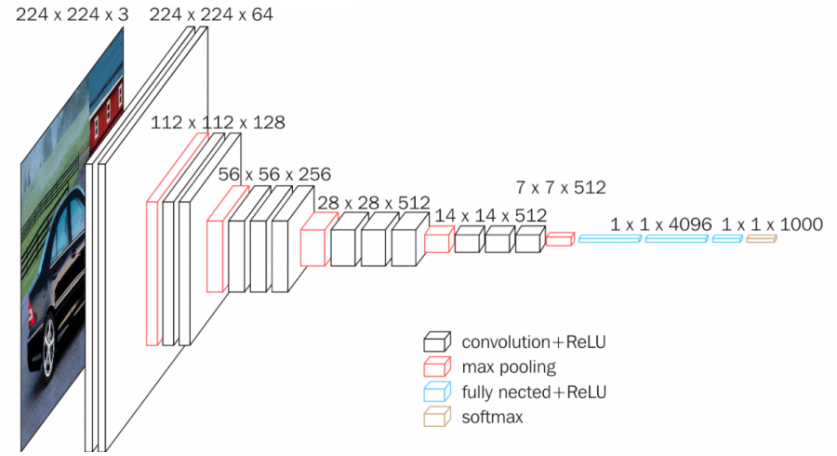

Figure 4. VGG16 – Convolutional Network for Classification and Detection

The top three fully connected layers are removed from the 16 layers of VGG16 to obtain an image of size (7, 7, 512). This image is then flattened to obtain an embedding of size 25088 for each image. To visualize the embeddings, t-SNE is used to plot them on both 2-dimensional and 3-dimensional vector spaces. t-SNE is a technique that can reduce high-dimensional data to low-dimensional data for visualization purposes while preserving the relationships between the data points.

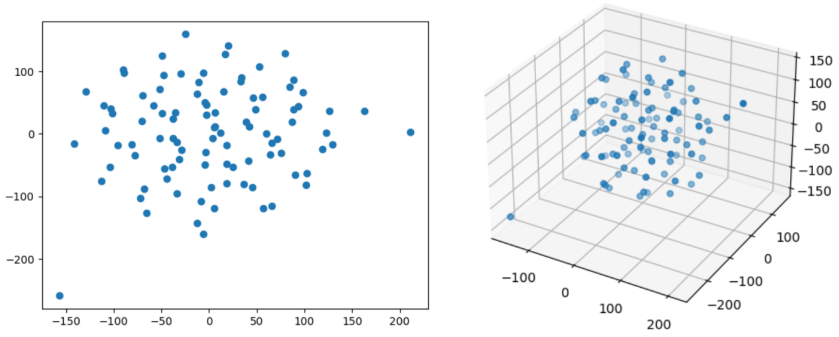

Figure 5. t-SNE to visualize VGG16 embeddings in 2-dimensional and 3-dimensional vector space

(c) **Model:** The training process involved splitting the data into training and testing sets using the train_test_split function from the scikit-learn library. The labels were encoded using label encoding. A logistic regression model was then trained on the preprocessed and embedded data using scikit-learn's LogisticRegression class. The trained model achieved an accuracy of 45%, indicating that the model's predictions were only slightly better than random chance. This suggests that more sophisticated models may be needed to accurately classify person behavior based on social media images and descriptions. Further experimentation with different models and hyperparameters could improve the accuracy of the classification.

3.2.3 Early Fusion Technique

The early fusion technique involves combining the image and text embeddings and passing them through a neural network. In this step, the image embeddings and text embeddings obtained in the previous steps were concatenated to create a single feature vector for each data instance. Then, the feature vectors were scaled using the MinMaxScaler and fed into a neural network with an input size of 25856 (i.e., 768 + 25088) neurons. The labels were one-hot encoded and the output layer of the neural network had 6 neurons to represent the 6 classes. The model was trained and tested on the concatenated embeddings and achieved an accuracy of 80% on the test data.

$$x_{scaled} = \frac{x - x_{min}}{x_{max} - x_{min}}$$

Figure 6. MinMaxScalar

3.2.4 Parallel Fusion using MLP(multi-layered perceptrons)

In this approach, the image and text embeddings are passed through separate neural networks, and the intermediate output of each network is obtained. These intermediate outputs are then concatenated and passed through another neural network layer for final classification.

The neural networks are optimized using the Adam optimizer, with the loss function being categorical cross-entropy. The metrics used for evaluation are accuracy. The activation function used in the intermediate layers is ReLU, while the final activation function used for classification is softmax.

$$\sigma(\vec{z})_i = \frac{e^{z_i}}{\sum_{j=1}^{K} e^{z_j}}$$

Figure 7. Softmax function

$$f(x) = max\,(0, x)$$

Figure 8. ReLU function

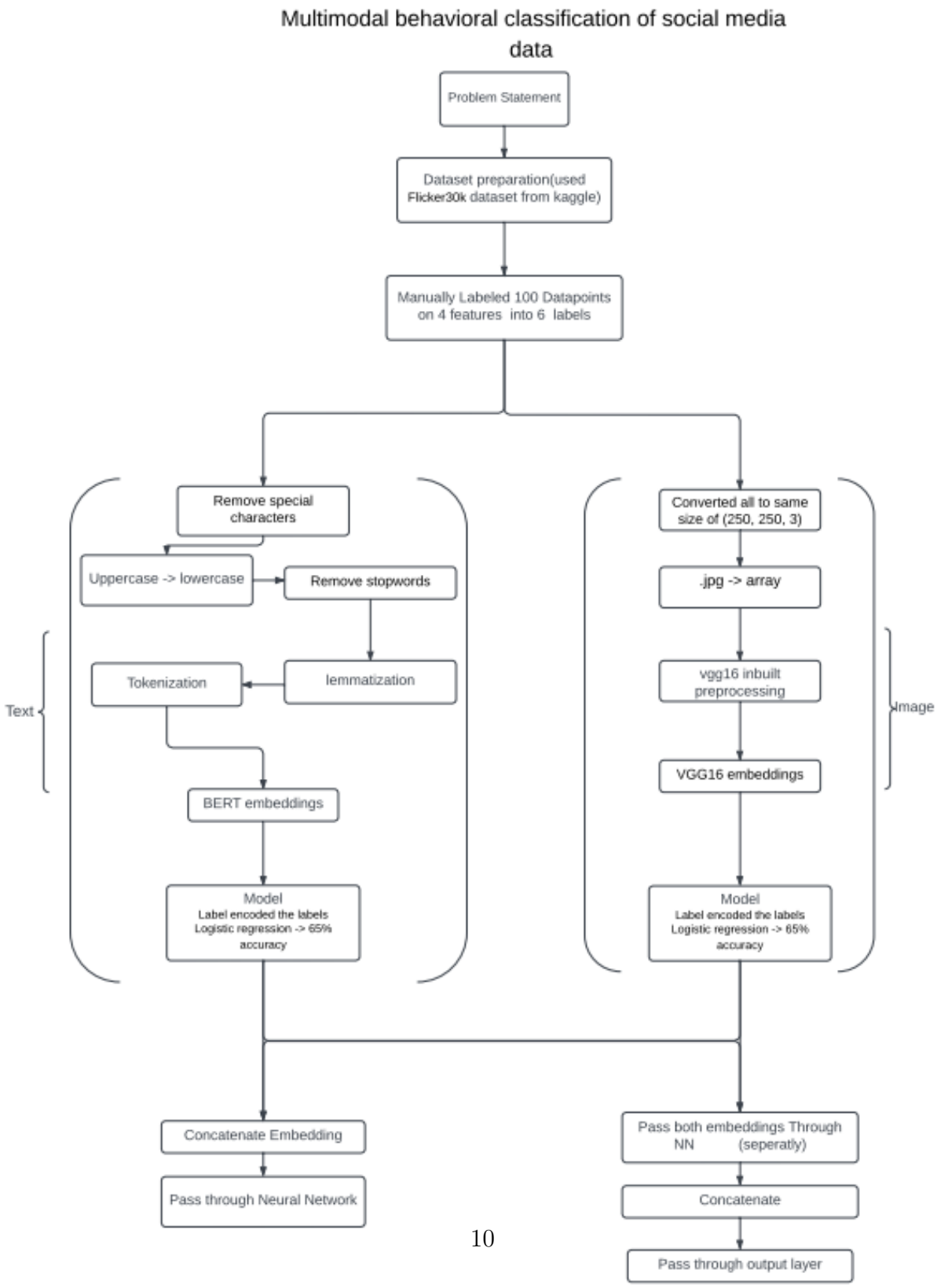

Figure 9. WorkFlow

4 Results

The models achieved varying degrees of accuracy. The logistic regression model on text data had an accuracy of 65%, while the VGG16 model on image data had an accuracy of 45%. However, when the image and text embeddings were concatenated and passed through a neural network, the accuracy improved to 80%.

```
model.fit(X_train, y_train, epochs=3, validation_data=(X_test, y_test))

Epoch 1/3
3/3 [==============================] - 2s 354ms/step - loss: 1.0875 - accuracy: 0.6447 - val_loss: 0.7245 - val_accuracy: 0.8000
Epoch 2/3
3/3 [==============================] - 0s 67ms/step - loss: 0.2409 - accuracy: 0.8553 - val_loss: 0.7159 - val_accuracy: 0.8000
Epoch 3/3
3/3 [==============================] - 0s 42ms/step - loss: 0.0761 - accuracy: 1.0000 - val_loss: 0.6947 - val_accuracy: 0.8000
<keras.callbacks.History at 0x7f2c63ff7220>
```

Figure 10. Final accuracy after tuning

Another approach of passing the image and text embeddings separately through neural networks and then concatenating the outputs achieved an accuracy of 75%.

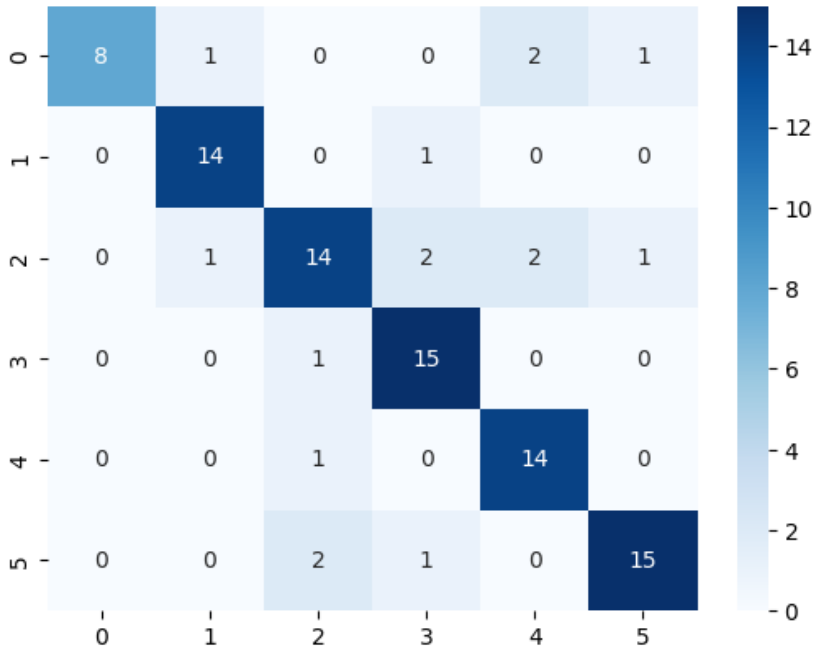

Figure 11. Confusion Matrix

Overall, the results demonstrate the usefulness of combining image and text embeddings in classification tasks and the importance of appropriate preprocessing and feature extraction techniques.

Instance	Precision	Recall	F1-Score	support
0	1.00	0.67	0.80	12
1	0.88	0.93	0.90	15
2	0.78	0.70	0.74	20
3	0.79	0.94	0.86	16
4	0.78	0.93	0.85	15
5	0.88	0.83	0.86	18
Accuracy			0.83	96
Macro-avg	0.85	0.83	0.83	96
Weighted-avg	0.84	0.83	0.83	96

Table 1
Classification Report

5 Conclusion

In this project, we explored different approaches to classify images and text data using machine learning techniques. We used two different datasets, one containing images and their descriptions, and the other containing text data related to different product categories. We applied preprocessing techniques to clean and transform the data, and used various visualization methods to better understand the data and the performance of the models.

For image classification, we used VGG16 embeddings to extract features from the images and built a neural network that achieved an accuracy of 45%. We then explored an approach where we concatenated the image and text embeddings and passed them into a neural network, which achieved an accuracy of 80%. We also experimented with passing the image and text embeddings separately into neural networks and concatenating their outputs in the middle, which yielded promising results.

For text classification, we used BERT embeddings to extract features from the text data and trained a logistic regression model that achieved an accuracy of 65%. We then concatenated the text and image embeddings to improve the classification accuracy of the model.

Overall, we demonstrated that combining multiple sources of information can improve the performance of machine learning models for image and text classification. Our project highlights the importance of preprocessing and visualizing the data before building the models, and the flexibility and power of using neural networks to extract and combine features from different sources. These techniques have wide applications in different industries such as e-commerce, healthcare, and finance.

References

[1] Emad Barsoum, Cha Zhang, Cristian Canton Ferrer, and Zhengyou Zhang. Training deep networks for facial expression recognition with crowd-sourced label distribution. In *Proceedings of the 18th ACM international conference on multimodal interaction*, pages 279–283, 2016.

[2] Divya Krishnani, Palaiahnakote Shivakumara, Tong Lu, Umapada Pal, and Raghavendra Ramachandra. Structure function based transform features for behavior-oriented social media image classification. In *Pattern Recognition: 5th Asian Conference, ACPR 2019, Auckland, New Zealand, November 26–29, 2019, Revised Selected Papers, Part I 5*, pages 594–608. Springer, 2020.

[3] Leqi Liu, Daniel Preotiuc-Pietro, Zahra Riahi Samani, Mohsen E Moghaddam, and Lyle Ungar. Analyzing personality through social media profile picture choice. In *Proceedings of the International AAAI Conference on Web and Social Media*, volume 10, pages 211–220, 2016.

[4] Dhara Mungra, Anjali Agrawal, Priyanka Sharma, Sudeep Tanwar, and Mohammad S Obaidat. Pratit: a cnn-based emotion recognition system using histogram equalization and data augmentation. *Multimedia Tools and Applications*, 79:2285–2307, 2020.

[5] Bryan A Plummer, Liwei Wang, Chris M Cervantes, Juan C Caicedo, Julia Hockenmaier, and Svetlana Lazebnik. Flickr30k entities: Collecting region-to-phrase correspondences for richer image-to-sentence models. In *Proceedings of the IEEE international conference on computer vision*, pages 2641–2649, 2015.

[6] Mukta Sharma, Anand Singh Jalal, and Aamir Khan. Emotion recognition using facial expression by fusing key points descriptor and texture features. *Multimedia Tools and Applications*, 78:16195–16219, 2019.

[7] Chandan Tiwari, Madasu Hanmandlu, and Shantaram Vasikarla. Suspicious face detection based on eye and other facial features movement monitoring. In *2015 IEEE Applied Imagery Pattern Recognition Workshop (AIPR)*, pages 1–8. IEEE, 2015.