Journal of Intelligent Computing System (JICS), Vol.1, Issue. 1, pp. 37-52, (Jan 2026)
ISSN: 3139-3616 (online)

# Human Mood Prediction Using Landscape Photographs with VGG16 and CuDNN-LSTM 5 folds Attention Algorithms

Anustup Mukherjee[1], Ankit Pandey[2], Sudha Velusamy[2]

[1]Machine Learning Engineer, Caimera
[2]Samsung Research Institute, Banglore, India
[1]anustup.mukherjee98@gmail.com, [2]ankit.pandey@samsung.com, [3]Sudha.v@samsung.com

**Abstract-** Mood's state depicts human personality traits and behavior in every situation. Predicting mood is a complex problem that deals with interactions of human psychology, subconscious mind thoughts, and their connection with the environment. This research proposes a deep learning-based solution to deal with such problems of the functional aspects of facial expressions and body gestures. Still, the scope of connecting psychology and the environment is yet unsolved. Mood prediction using landscape photographs is a unique computer vision problem that solves abstract image analysis issues. Human moods are always dynamic and dependent on the surroundings and their interactions. Moods can easily predict the relevance and requirement of any situation and its improvements, though very few computer vision architectures have solved this problem in a detailed and accurate manner. This work uses a unique and robust architecture of Image captioning with sentiment analysis to solve the problem correctly and production-ready. To analyze the model performance, 7440 landscape pictures were collected from different sources mentioned in the dataset section. After the data cleaning and annotation, 6912 landscape pictures for Image captioning were sent to the model, eliminating 528 images due to duplicates. As a result, VGG 16 LSTM trained on an unseen dataset, which attained training and testing accuracy of 94.5% and 87.9%, respectively. Further, CuDNN LSTM (5-fold attention) is used for the sentiment analysis on the same dataset and achieved training and testing accuracy of 93% and 81.1%, respectively.

**Keywords**: *Landscape Images, Computer Vision, CNN, MobileNet v2, VGG 16, LSTM, Image Captioning, Sentiment Analysis, CuDNN, BERT.*

## 1. Introduction

The interaction between human psychology and nature defines human moods. Moods are dynamic and depend upon factors like facial expressions, hormones, and environmental exposure. Findings suggest that an individual feels significantly better when exposed to nature scenes [1] . According to [2], outdoor visual environments are expected to influence psychological well-being. The extent to which an individual has been visually exposed to nature impacts one's feelings and has positive or negative effects on their well-being. Experimental studies by the National Library of Medicine prove a direct relationship between mood and a landscape [3]. Humans have psychological disorders such as mood swings, depression, and lack of emotion. To depict individual moods or emotions is tedious. Humans habitually use filter images to express their feelings. In respect of this, a landscape picture can be used where filters are added that make it livelier and more interactive. However, a person's mood varies according to their vicinity around the individuals and thoughts. According to [4], [5], [6], the globe has introduced theories and logic for predicting a person's mood based on multiple factors, including facial expression, body gestures, tone of voice, attitude, etc. Social media, like Facebook, allows people to get recommendations based on their mood. In telemedicine, a doctor organizes a psychological session to predict a patient's behavior using their mood and expressions. Computer Vision (CV) contributes to predicting human moods with the help of body gestures and facial expressions using neural network approaches implemented to understand mood features based on human attributes. However, the major problem is "How humans pretend to their mood by their behavior?"

CV algorithms still struggle to understand human-to-nature interaction or how people's surroundings impact their moods. When we deal with mood detection from facial expressions, the expression matrices are easily detectable parameters that make an AI model learn quickly. In the case of landscapes, several constraints must be considered while executing a mood detection engine, such as object-specific mood on landscapes. Suppose a person is standing in front of the Taj Mahal despite any weather or other geographical, mental, and physical conditions. In that case, at least the person will feel happy, surprised, or joyful. Hence, objects with this type of special significance lead to special attention when detecting a landscape mood [7]. The color-specific mood on landscapes is the most important distinguishing factor for mood detection from real-time scenarios, i.e., when the



© 2026 JICS. Personal use is permitted, but republication/redistribution requires JICS permission.
see: https://synergyworldpress.com/journal/jics/about/overview

*doi.org/10.67420/109319.1.1.3*

weather is rainy, it will have a blurry dark color that can be categorized as sad [8]. While considering a sunshine landscape that is indeed a bright texture, it can be classified as happy. Colors play an essential role in making models learn about the emotional perspective of the landscapes. The complexity arises in this domain when multi-emotion semantics on the same colors are considered. For example, suppose everyone feels scared or angry after witnessing a fire, when a model is trained on the color aspect. In that case, the model learns the yellow color under the training label fear and anger, which will point to these classes.

On the other hand, there is a picture of spring where the yellow or orange leaves of trees blossom everywhere, which can be categorized as happy and cherishing. In this context, the model will get confused, resulting in a comfortable or fearful emotion-specific mood on Landscapesis, the most challenging problem one can think of while predicting the mood. In regular mood detections, face, text, and emotions can be understood from expressions or keywords in writing. But for a landscape, it's all about what the brain thinks. A person may feel surprised after seeing a sunset, whereas another person under stress may feel sad after seeing it on the same evening. So, it depends on the mindset and the nature of psychology involved behind the landscape and mood. The model needs a proper understanding of the standard psychology behind the scenes involving the mixed judgment of objects and multi-colors to detect these emotions. In [9], proposed the MldrNet model to classify the image emotions by varying the number of convolutional layers. They concluded that image emotions are not only affecting the high-level semantics of the image. However, it affects an image's middle and lower-level visual features such as texture, color, etc. VGG 16, developed by stacking multiple convolution and pooling layers, is a backbone network for effectively extracting low-level visual features [10], [11].

Based on the above analysis, it has been concluded that no such model is trained that classifies the emotion accurately by seeing a photograph. The proposed work focuses on predicting an expected human mood based on nature's context (i.e., mood prediction using landscape images). Though, the problem is abstract with a bunch of unsupervised cases. There is no defined pattern for understanding nature or psychology, so generating an inference for prediction at a full scale is challenging. It is a complex problem as it encounters deep-end complexities regarding color psychology. For example, the moods of different persons may vary after witnessing a sunset image. One might feel happy, but it's not necessary that the other person feels the same. So, this problem is typically dynamic for the viewer's different color and mood conditions. In this work, a customized AI algorithm generates a descriptive caption for any landscape image based on the image's concept and context. A sentiment analyzer further analyzes the conceptual caption to detect a human mood based on inter-word relationship understanding.

This paper is further divided into different sections: Section 2 discusses the detailed description of dataset collection and processing. After this, the proposed model with the mathematical formulation is covered, and section 3 discusses the experimentation formulation using different techniques. Finally, the paper is concluded in section 4 with its future aspects.

## 2. Materials and Methods

The following sections contain details about the dataset and the formulated model architecture of the proposed algorithm.

### 2.1. Dataset

The dataset of this work has been collected from three different sources. The first dataset has been collected from the Kaggle repository of Landscape Image source [12], which contains 4319 landscape images. The second dataset was collected from Geo Pose 3k Landscape Imagery [13], which includes 1500 landscape images; the third dataset was collected from Landscape Imagery Unsplash [14], which contains 1621 landscape images. The collected dataset consists of 7,440 landscape images and is used for predicting the mood of human beings. This data is further divided into training (i.e., 60%) and testing (i.e., 40%) of collected landscape images. In addition, a separate set of unseen images (i.e., 140 Landscape images) on which models' production accuracy is tested.

### 2.2. Pre-processing of Dataset

In the model formulation, a complete docker container is set up with Hugging Face Context Image Captioning Tool [15] that uses Image GPT [16] and Vision Transformers [17] to generate conceptual and context-aware captions. After this, the entire dataset is passed to this model using an automated python script to create and store the captions in a CSV file. The complete captions were generated to formulate the captioned dataset's categorization. The categorization of images was done using Freud's text psychology [18] and real-time context-based emotion analysis to provide an emotion for the corresponding caption. The emotions considered are Happy, Sad, Anger, Fear, Disgust, Surprise, and Neutral. Finally, the Boyer-Moore Maximal Voting algorithm [19] is applied in the CSV dataset to sort out the best-fit emotion label for the caption.

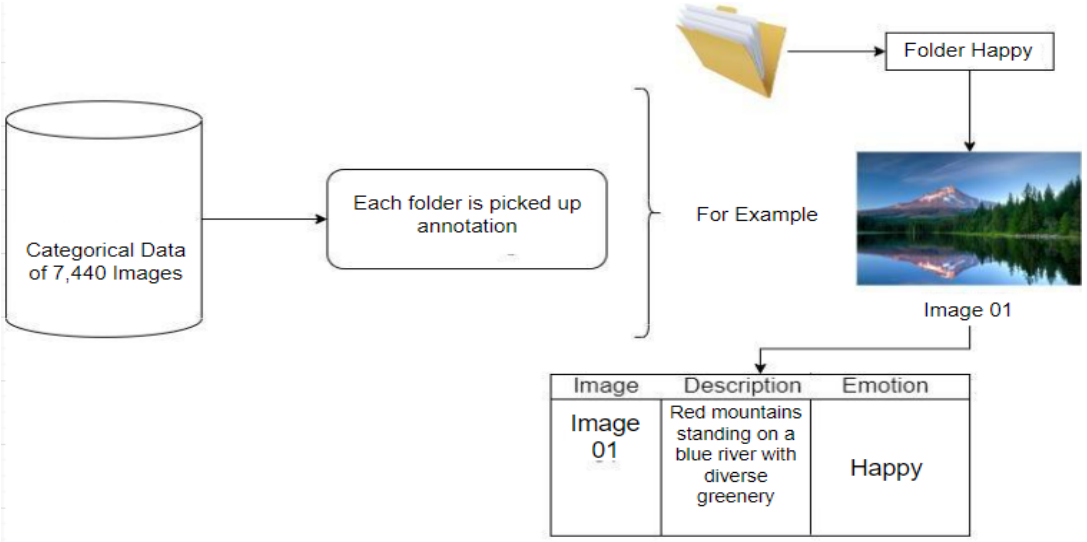



**Figure 1.** Complete method and steps of data collection and processing for mood prediction using landscapes

In this algorithm, each row is considered an array with labels as elements. The maximal repeating element is used as the final label of the caption. When no repeating element is found, the label is set to neutral.

Similarly, images are eliminated from the dataset when two elements have the exact repetition count and are labeled as confusion classes. 528 images are deleted from the collected dataset due to duplicate. After the algorithm implementation, each caption is marked with a maximally voted label. Figure 1 shows the complete process of data collection and annotation.

### 2.3. Proposed Model Formulation

A customized approach (as shown in Figure 2) to predict the mood has been developed using context and concept-aware image captioning with sentiment analysis. Firstly, an image captioning model is trained on the collected dataset using VGG 16 LSTM residual architecture. The model is then modified by writing a custom feature extractor script that applies Freud's color psychology analysis and context judgment. The script separately generates feature maps for the background as a context and significant objects. The image captioning model gives conceptual captions as an output. It is stored in a testing dataset for the final testing of moods from unseen captions. Then, a CuDNN LSTM with a 5-fold attention model for sentiment analysis is trained using sentiment and caption data. Global attention is set up for position vector analysis. The model is specifically customized to understand the context of the words with the sentiments and the inter-word relationship between words with close meaning. The multi-level abstraction using attention folds is implied. Each layer/hop retrieves important context words and then transforms the representation received from the previous level to the next slightly more abstract level. The proposed model is able to learn with the composition of such transformations and very complex functions of sentence representation towards an aspect.

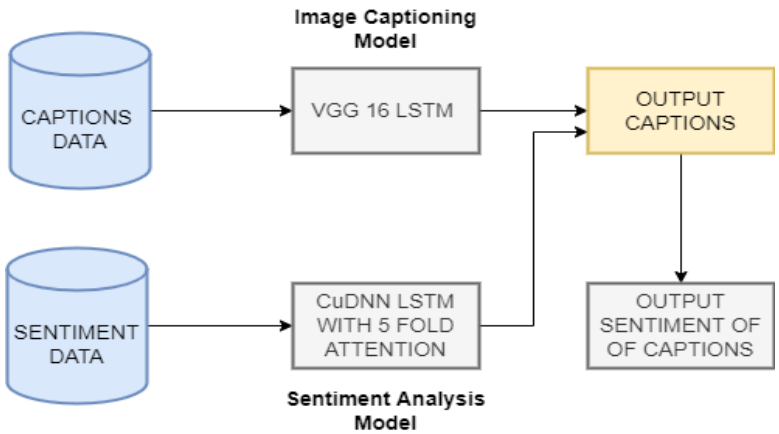



**Figure 2.** Algorithm flow to detect the mood of landscape images using image captioning and sentiment analysis

The image captioning and sentiment analysis model formulation is discussed as mentioned below:

**Step 1: Feature Extraction setup for image captioning**

This step prepares a custom feature extractor script to make the captioning model context and concept aware. The features extracted in the network are based on differentiation by parts mechanism. Let the input image be i, which has total pixel calculation (as shown in eq. (1) and (2))

$$P_i = (p_1, p_2, p_3, p_4, \ldots., p_m) \quad (1)$$

$$\left[P_i = \sum_{i=1}^{m} P\right] \quad (2)$$

proposing a method to extract features from the image using a differentiation-by-parts mechanism, where the pixel values are used to create collaborative clusters of features through differential equations. This approach can capture context and concepts within the image for improved image captioning. For this, the model will use differential equations on eq. (3) to distribute the feature section into collaborative clusters where the individual features represented by f are differentiated with respect to p, which results in the total feature matrix represented by F.

$$\frac{d(f)}{d_{p_i}} = \{f_1, f_2, f_3, f_4, \ldots., f_m\} = \sum_{i=1}^{m} F \quad (3)$$

These clusters, each an ensemble of individual features 'f,' embody the epitome of contextual and conceptual fusion, poised to redefine image captioning's depth and precision. After differentiation from the eq.(3), the features set is extracted, and the features with similar matrix configurations or similar probabilistic matrix are mapped into one set in eq. (4).

$$P\,(f) = p(f_1, f_2, f_3) + p(f_4, f_5, f_6) + \cdots + p(n) \quad (4)$$
$$= p\left(\sum_{i=1}^{n} F\right)$$

and, $[m_1 = m_2]$ $\begin{cases} m_1[f_1] \in [1 \ldots \ldots N] \\ m_2[f_2] \in [1 \ldots \ldots N] \end{cases}$

Where p, m, and n represent probability, matrix size, and the number of elements, respectively. A group feature set in eq. (5) can be obtained:

$$\mathrm{F} = f_1' + f_2' + \cdots + f_n' = \sum_{i=1}^{n} f_n \quad (5)$$

Now, the total integration is created by using the integration function $[\int F_N' = \sum F_N] = 'f$ passed to the network for further processing.

**Step 2: VGG 16 LSTM for Image Captioning**

This model uses the custom feature extractor, as mentioned above script (i.e., step 1) to get trained on the dataset. Figure 3 shows the following modifications in the algorithm:

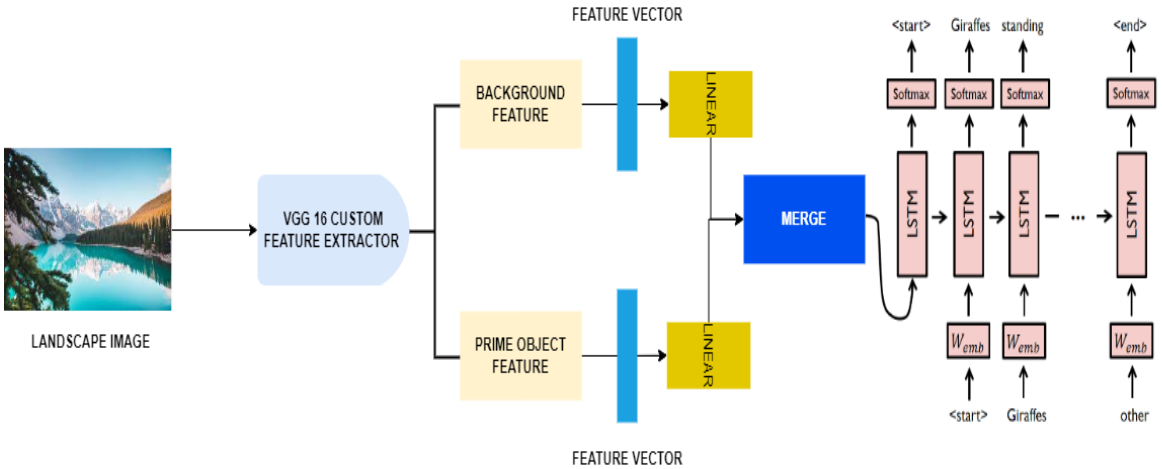



**Figure 3.** VGG 16 LSTM for Image Captioning

**Step 2.1. VGG 16 Model setup**

For image captioning, let 'f' denote the feature vector which is extracted from the image generated from the VGG 16 model, and S = $\{s_1, s_2, s_3, s_4, \ldots, s_m\}$ be the reference sentence of the image and $s_i$ is the i[th] word, the target is to approximate the distribution of sentence as: $\left(\frac{s}{f}\right)$

By applying the Bayesian theorem chain rule on $\left(\frac{s}{f}\right)$ the log distribution is represented in a joint distribution form (as shown in eq. 6)

$$\left(\frac{S}{f}\right) = \sum_{t=1}^{N} (s_t | f, s_1, .., t-1) \quad (6)$$

Where each word $S_i$ in S is represented in the form of words embedding and $s_1, .., t-1 = \{s_1, s_2, \ldots, s_{t-1}\}$ is the slice of 1 to t-1 element of S.

To model from eq.(6), $(s_t | f, s_1, .., t-1)$ RNN is employed to calculate the output using eq. (7).

$$h_t = F(s_t, h_{t-1}) \quad (7)$$

Where F denotes RNN cell, $(h_{t-1})$ represent the output of (t-1) time stamp. RNN can have sequence information in sentences, but due to the vanishing gradient problem, it cannot attain long-term dependency. LSTM, the RNN variant, can remember information over a long period. The three different gates, namely the forget gate, input gate, and output gate, control the memory which is the core of LSTM, and store the information of each time step. These gates are guided by $h_{t-1}$, $s_t$ respectively, by the f-cell function. Therefore, to build a proposed language module, LSTM as the basic module is employed. At every time step, each word embedded feature, i.e., $s_i$ is input to the language model. Afterward, all the information that is present in a hidden state is estimated using $h_t$. Lastly, to predict the $s_{t+1}$ distribution, the SoftMax activation function is employed. At the beginning of every sentence, a special word called Start of Sentence (SoS) is added for training convenience. Finally, for the backpropagation, the cross-entropy loss function is implemented. The sentence having the maximum confidence score is the final image captioning winner.

**Step 2.2 Residual Architecture Formulation**

Generally, one or two RNN or LSTM layers are employed in the language shallow model. Due to this, the representation and discrimination of the language features are shifted toward the degrading side. Further, the vanishing gradient problem usually occurs in multi-layer LSTM networks like Stacked LSTM. This phenomenon is also found in [20] , concluding that the model's best performance was achieved in two-layered stacked LSTM, while the model's performance is degraded if more than two LSTM layers are employed. In this article, a residual network architecture is proposed. First, let $g^0$ stands for the LSTM model. The output of $g^0$ depends on $s_t$ and hidden state ${h_{t-1}}^0$ is shown in eq. (8).

$${h_t}^0 = g^0({s_t}, {h_{t-1}}^0) \quad (8)$$

Inspired by residual architecture, another shallow LSTM model $g^1$ is used to fit the residual function with the output $g^0$ and ${h_{t-1}}^1$as shown in eq. (9)

$${h_t}^1 = f(s_t, h_{t-1}) - g^0({s_t}, {h_{t-1}}^0) = g^1({h_t}^0, {h_{t-1}}^1) = g^1(g^0({s_t}, {h_{t-1}}^0), {h_{t-1}}^1) \quad (9)$$

As $g^1({h_t}^0, {h_{t-1}}^1)$ + $g^0({s_t}, {h_{t-1}}^0)$, which forms a new approximate function as $f(s_t, h_{t-1})$

The residual approximation step can be employed in a recursive manner of computation. The general form to derive the residual approximation function as the 1[st] layer is shown in eq. (10)

$$g^l({h_t}^{l-1}, {h_{t-1}}^l) = f(s_t, h_{t-1}) - g^0({s_t}, {h_{t-1}}^0) - \sum_{t=1}^{l-1} g^l({h_t}^{l-1}, {h_{t-1}}^l) \quad (10)$$

**Step 2.2.1. Temporal dropout**

As the number of layers in stacked LSTM increases, it may cause model overfitting, especially when working with a small-scale dataset. Dropout helps to avoid the overfitting situation and regularize the model by randomly discarding some neurons from the hidden layers. The suggested VGG_LSTM model employs temporal dropout in each LSTM layer between two-time steps. Adopting this, it randomly drops some connection in batch from the hidden state, e.g., $h_{t-1}$.

Further, LSTM is used to capture and model long-range dependencies in sequences using forget, input, and output gates, which enables them to retain certain information about language patterns over an extended context. Finally, time step t (fed into the hidden state) is updated as presented in eq. (11)

$$r_{t-1} \sim Bernoulli(p),$$

$$h'_{t-1} = h_{t-1} \odot r_{t-1} \quad (11)$$

Where $Bernoulli(p)$ generates a vector of independent Bernoulli random variables, each of which has the probability $p$ of being 1, . $h_{t-1}$ and $h'_{t-1}$ represent the original hidden state and the updated hidden state at the time step of $t-1$, and $\odot$ means the element-wise multiplication operation.

**Step 2.3 Sentiment Analysis for CuDNN LSTM with 5 folds of Attention**

The output caption of the above-proposed model VGG 16 LSTM is passed for further sentiment analysis using CuDNN LSTM with a 5-fold attention network (as shown in Figure 4).

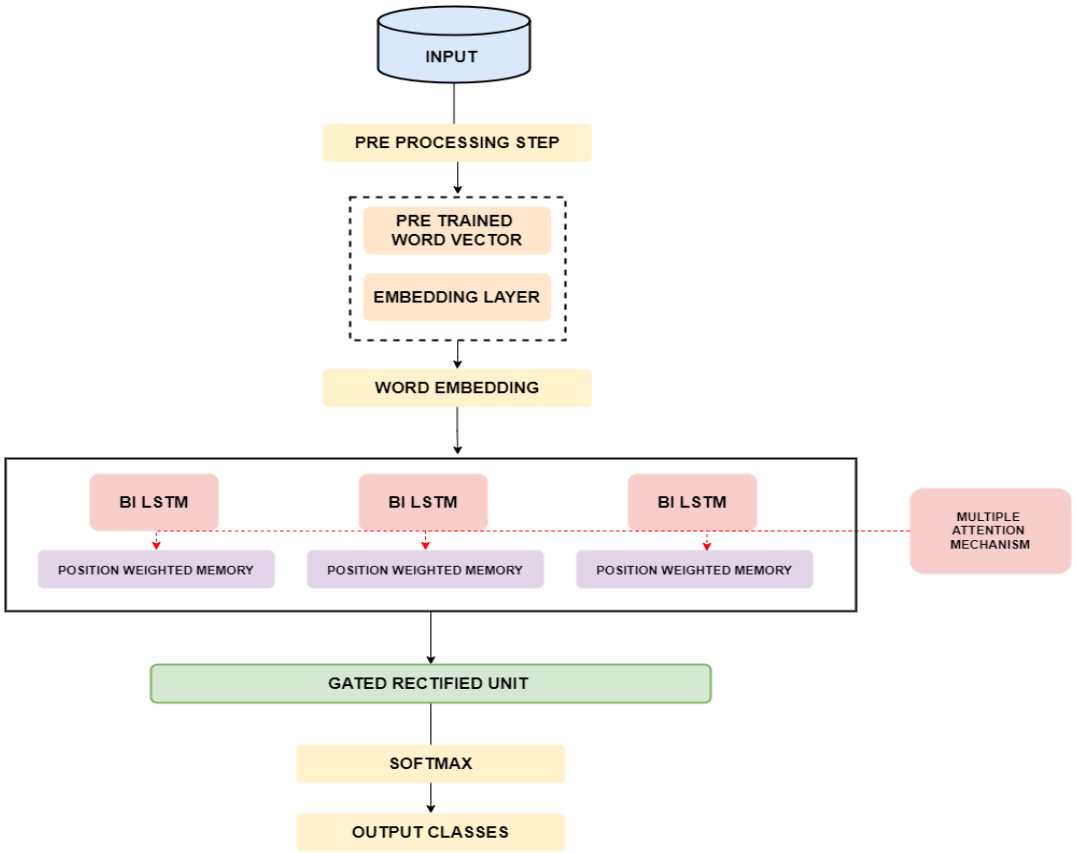



**Figure 4.** Sentiment Analysis using CuDNN LSTM 5-fold attention

This approach uses an attention mechanism that first fixes the position weight indexes of the individual word vectors, where the assignment of the weights is pre-determined and does not change during attention mechanism operations. By fixing the position weight index of the word vector, the attention mechanism is designed to place more emphasis on certain words or positions within the sequence. Further, positions weighted memory plays a crucial role in deciding where to apply the attention. In this, the position-weighted memory refers to a weighted representation of the memory content based on the pre-defined weight indexes assigned to each position in the sequence. This implies that the position-weighted memory, which reflects the importance of words based on their positions, is a critical factor in determining where the attention mechanism should focus on it. The target word is first identified, and the words close to the target word are scored relatively high compared to those far off.

Further, GRU updates the hidden states/ weights after each attention by focusing on different parts of the input sequence, as highlighted by the attention mechanism. A better prediction of sentiments is achieved by non-linearly combining the results from multiple attentions using a gated recurrent unit. Multiple attention layers with diverse abstraction levels allow learning the text representation by deep memory networks. Every hope retrieves important context word information and then transforms the representation to an abstract level received from the last level. Finally, the complex representation of sentence function is learned via such compositional transformation [21].

The data is standardized by the matrix dimensionality reduction theorem [22], which follows the 'NN' model input format. Now, based on the image dimension, the total possible feature set is calculated as shown in eq.(12)

$$n_dim_{P_i} = n_{featureset} \quad (12)$$

All the possible features of the combinatorics matrix are obtained by performing the above permutation. The features are concatenated into a single matrix M, and the matrix is further passed into the CuDNN model.

A serialized LSTM architecture is followed to understand the data position and activate the signal in the following layers. The features are transformed into a series by using mathematical computations:
Let's consider the feature matrix to be M and a set of features to be 'f' then:

$$m(f) = f(\triangle A + h) \quad (13)$$

In eq.(13), 'A' represents the linear transformation between 'm' to a series. The quantity is minimized, as shown in eq.(14)

$$'s' = \sum_{x=1}^{t} [f(\triangle A + h) - m(f)]^2 \quad (14)$$

To determine the shift of 'A' adjustment in eq. (14) following modifications are imposed as shown in eq.(15):

$$f(f(A + \triangle A) + (h + \triangle h) \approx f(\triangle A + h) + (h \triangle A + \triangle h) \frac{\partial}{\partial f}(F(x)) \quad (15)$$

After imposing all the series in eq. (15), the model LSTM will read the series in the bi-directional pattern. Attention folds are implied to avoid the problem of overfitting in the patterned training phase, as shown in eq. (16):

$$Attn\,(Q, k, v) = Softmax\left(\frac{Qk^t}{\sqrt{d_k}}\right)v \quad (16)$$

**2.4. Performance Metrics**

The performance metrics that are used to analyze the model performances are True Positive (TP), True Negative (TN), False Positive (FP), and False Negative (FN). Based on these, the calculation of precision, recall, accuracy, F1 score, and sensitivity for the image captioning model is accomplished. (as shown in eq.(17-21)).

$$Precision = \frac{TP}{(TP+FP)} \quad (17)$$

$$Recall = \frac{TP}{(TP+FN)} \quad (18)$$

$$F1 = 2 * \frac{Precision*Recall}{(Precision+Recall)} \quad (19)$$

$$Accuracy = \frac{TP + TN}{(TP + TN + FP + FN)} \quad (20)$$

$$Specificity = \frac{TN}{(TN+FP)} \quad (21)$$

For analyzing the sentiment analysis model, the Matthews Correlation Coefficient (MCC) (eq. (22)) is used.

$$MCC = \frac{(TP * TN) - (FP * FN)}{\sqrt{(TP + FP)\,(TP + FN)\,(TN + FP)\,(TN + FN)}} \quad (22)$$

Now, to analyze the actual performance of the captioning models, the BLEU score is calculated for the individual outputs (as shown in eq.(23))

Base Prediction (BP) = $\begin{cases} 1 & if\ c \geq r \\ e^{\left(1-\frac{r}{c}\right)} & if\ c \leq r \end{cases}$

$$\text{BLEU} = BP * exp\,(\sum_{n=1}^{N} w_n\ log\ log\ p_n) \quad (23)$$

$$= e^{min}\left(1 - \frac{len(reference)}{len(prediction)}, 0\right)$$

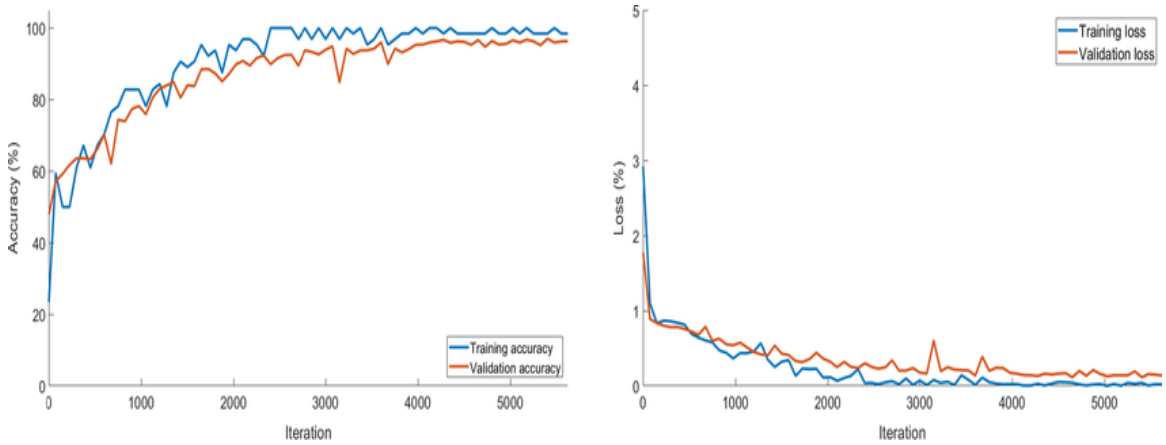
All the generated outputs are cross-validated with the ground truth values to understand the model's production accuracy.

## 3. Experimental Result Analysis

Experiments are conducted to modify and customize the proposed architecture for existing models and State-of-the-Art (SOTA) methods. The experiments are separately carried out for both approaches, respectively. Each experimental model is trained on a collected dataset and cross-tallied with the proposed architecture in terms of performance metrics. The model used for image captioning, i.e., VGG 16 LSTM, is trained on 5500 epochs with a learning rate 1e-5 using Adam optimizer. The CUDNN LSTM model is trained on 100 epochs with a learning rate of 1e-3 using the Adam optimizer. AMD RYZEN 1660 and NVIDIA 155K6 Ti GPUs were used to train the models.

### 3.1. Image Captioning

The proposed custom VGG 16 LSTM for context and concept-aware image captioning has experimented with some SOTA methods and existing approaches. The resultant accuracy plot of VGG 16 LSTM is shown in Figure 5(a), the loss plot in Figure 5(b), and the training and validation accuracies are mentioned in Table 1. The experimentation started with feature extractors in the CNN family extending to the Show Attend & Tell model, VGG 16 attention, and Noisy Student.




(a) (b)

**Figure 5. (a)** Accuracy plot of VGG 16 LSTM **(b)** Loss plot of VGG 16 LSTM

#### 3.1.1. CNN Family- Feature Extractors LSTM

Various experiments are carried out with Inception Net LSTM and Mobile Net LSTM. The experiment architecture for both the models involves Inception as a feature extractor with a GlovE word Embedding RNN for the word library connected with the LSTM to get the captions [23]. The concept behind using the embeddings is it has pre-trained word vectors, which will reduce the word-to-word relationship problem while framing the conceptual sequence. Apart from this, the custom feature extractor script used for VGG 16 LSTM is used in the experiment. The resultant accuracy plot of InceptionNet LSTM and MobileNet LSTM is shown in Figures 6 (a) and 7(a), the loss plot in Figures 6 (b) and 7(b), and the training and validation accuracies are mentioned in Table 1.

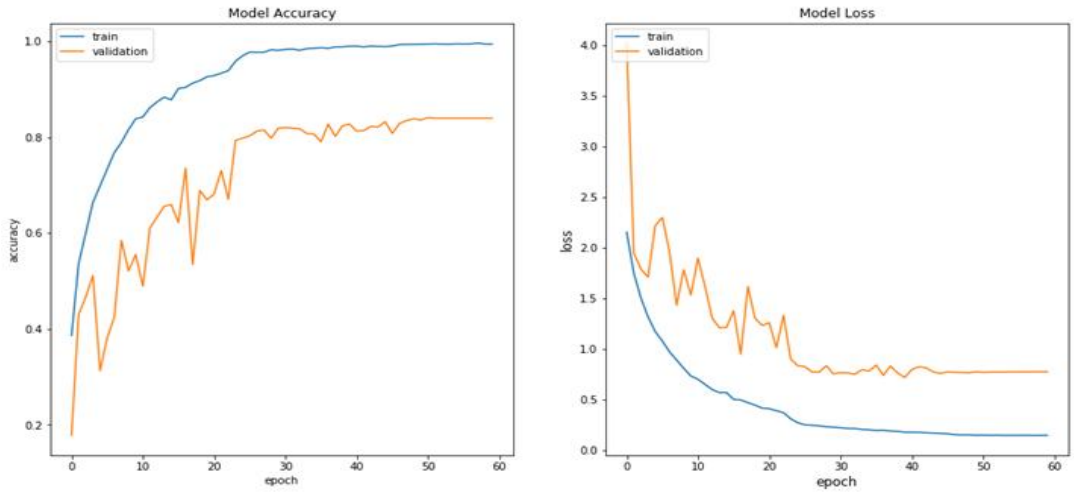



**Figure 6. (a)** Accuracy plot of InceptionNet LSTM **(b)** Loss plot of IncpetionNet LSTM

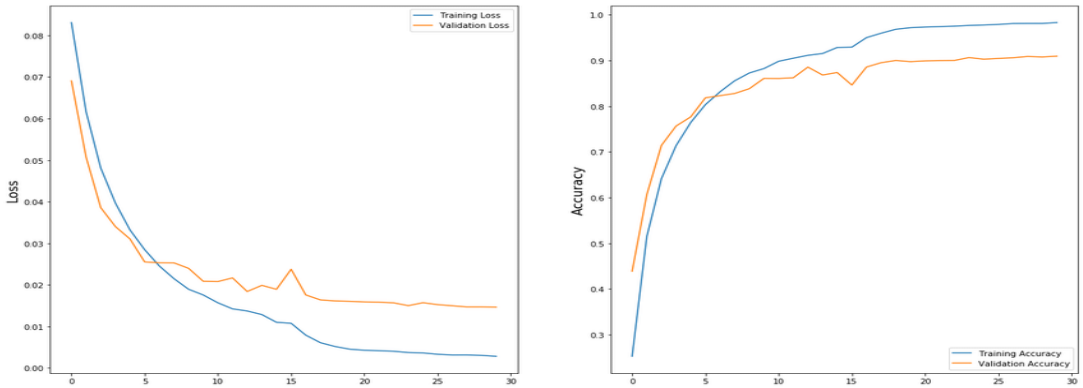



**Figure 7. (a)** Accuracy plot of MobileNet LSTM **(b)** Loss plot of MobileNet LSTM

### 3.1.2. VGG-16 Attention

Experimentation by adding attention mechanisms to existing VGG 16 architectures is made to observe the change in accuracy for adding concepts of context vectors. Global attention is involved in this experiment to consider all hidden state conditions and to derive the context vector. The resultant accuracy plot of VGG 16 attention is shown in Figure 8 (a), loss plot in Figure 8 (b), and the training and validation accuracies are mentioned in Table 1.

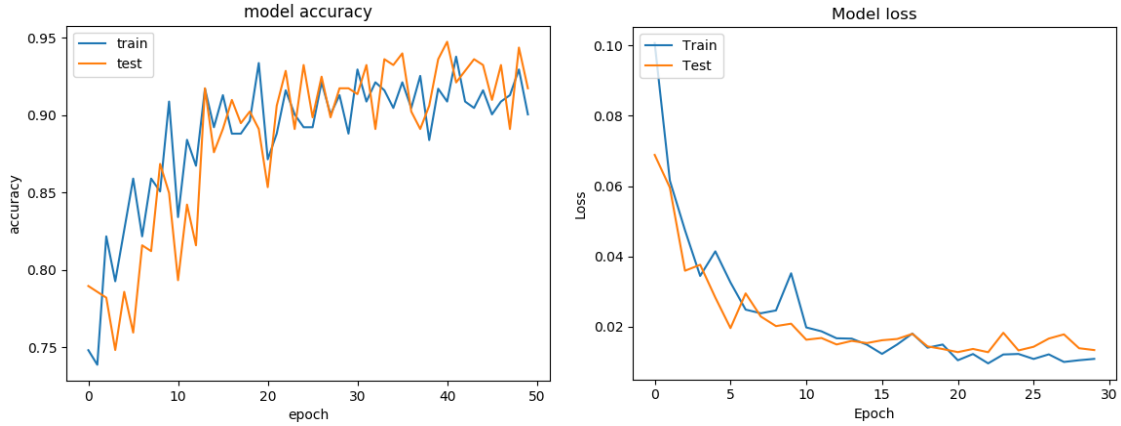



**Figure 8. (a)** Accuracy plot of VGG 16 attention **(b)** Loss plot of VGG 16 attention

### 3.1.3 SHOW ATTEND & TELL and Noisy Student

Show Attend & Tellis is a neural image captioning algorithm that uses Visual attention. It considers all the lower and upper bounds of the image and generates a conceptual caption. It is marked as a SOTA model in conceptual image captioning. The experimentation is carried out to find how much our architecture reaches the SOTA. The resultant accuracy plot of SHOW ATTEND & TELL is shown in Figure 9 (a)) the loss plot in Figure 9 (b), and the training and validation accuracies are mentioned in Table 1.

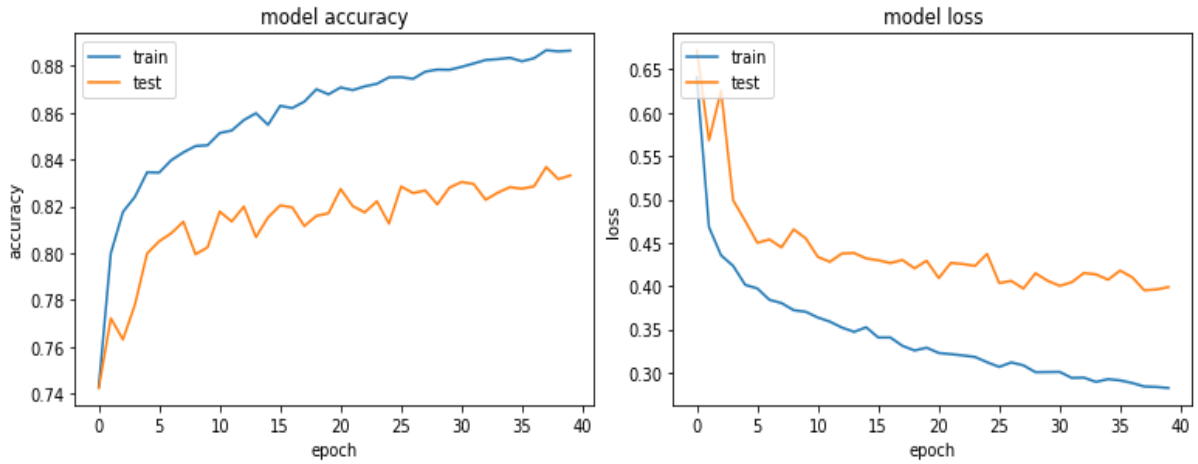



**Figure 9. (a)** Accuracy plot of SHOW ATTEND & TELL **(b)** Loss plot of SHOW ATTEND & TELL

Experimentation with Noisy Students was carried out as it is a semi-supervised algorithm. It is implemented to determine the scope of proposed approach in a supervised state. The resultant accuracy plot of Noisy Student is shown in Figure 10 (a), loss plot in Figure 10 (b), and the training and validation accuracies are mentioned in Table 1.

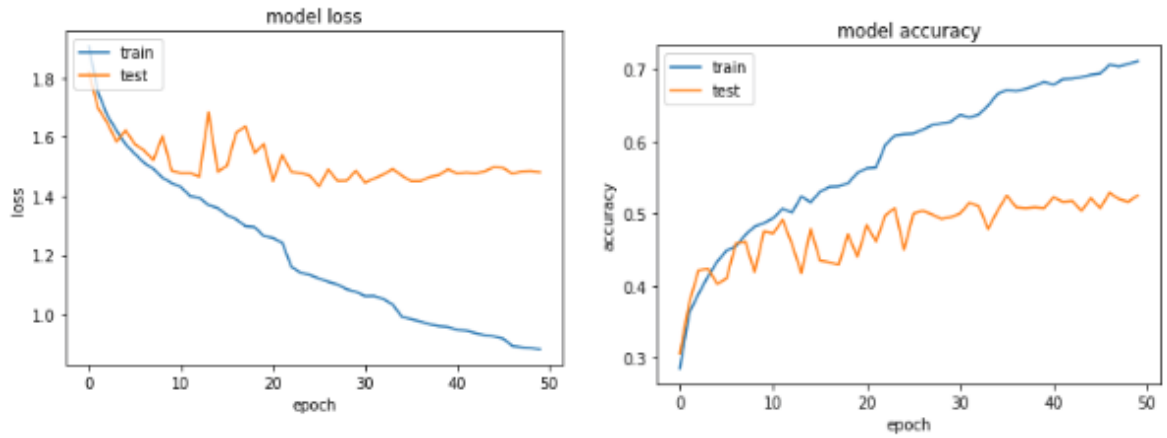



**Figure 10. (a)** Accuracy plot of Noisy Student **(b)** Loss plot of Noisy Student

### 3.2. Sentiment Analysis

After experiments on the image Captioning model, a set of experiments is carried out on existing SOTA NLP models for sentiment analysis. It includes experimentation with the BERT Family, SVM, Word2Vector model, and XL Net.

#### 3.2.1 Support Vector Machine (SVM)

The SVM is implemented using the TF-IDF vectorizer technique, where each word is vectorized and passed for processing. A simple classification concept using hyperplanes in SVM is used to train. The sentiment data, with columns of captions and sentiments, is given to the models, which report poor precision values and less testing accuracy. The confusion matrix of the SVM training is shown in Figure 11.

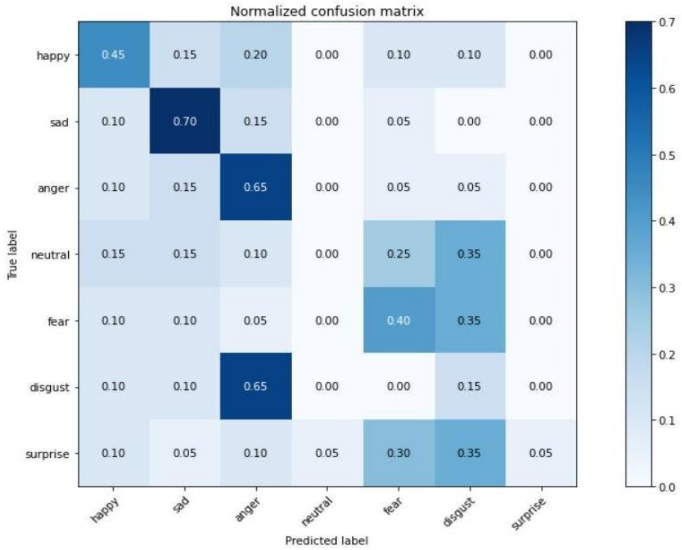



**Figure 11.** Confusion matrix of SVM

#### 3.2.2. LSTM Word2Vector

The primary reason for this experiment after experimenting with SVM, as the problem pointed out above, is that the simple classification architecture does not understand the words and the context relationship. LSTM word2Vec uses word embeddings and an LSTM block to learn over the units. The specialty of this model is a vector space of several dimensions produced by Word2Vec. In this, the word having a common context is placed near to each other in the space with unique words. It is employed in two ways: from a single word to predict its context (Skip-gram) or from the context to indicate a word (*Continuous Bag-of-Words*). The resultant confusion matrix of LSTM

W2V is shown in Figure 12, training and validation accuracies are mentioned in Table 1, and the complete performance metrics are updated in Table2.

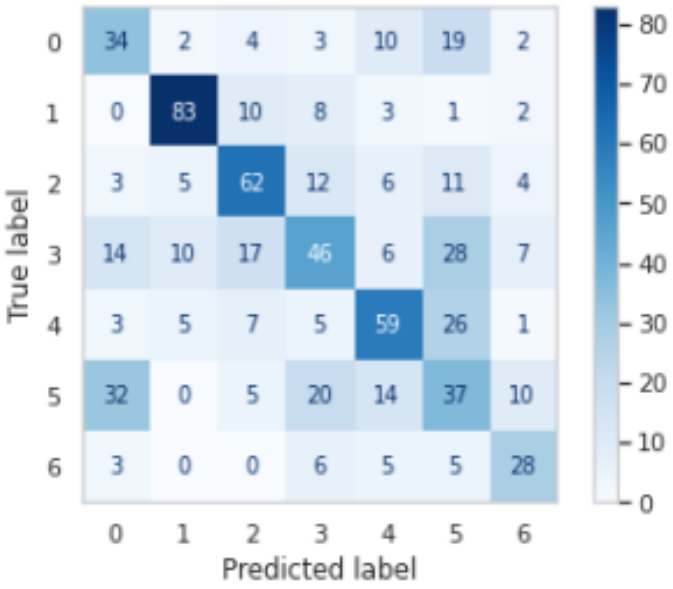



**Figure 12.** Confusion matrix of LSTM W2V

### 3.2.3. BERT (Fast AI)

BERT is a SOTA model in sentiment analysis, and it uses contextual word embeddings to understand the feature relationships. The primary goal of this experiment is to find out how close the approach reaches the SOTA. The resultant confusion matrix of BERT(FAST AI) is shown in Figure 13, training and validation accuracies are mentioned in Table 1, and the complete performance metrics are updated in Table 2.

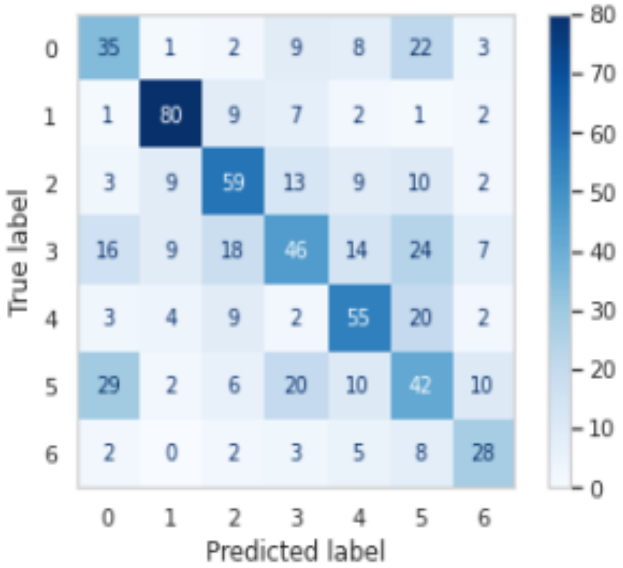



**Figure 13.** Confusion matrix of BERT (Fast AI)

### 3.2.4. DISTILL-BERT

It retains 97% performance using a BERT version to learn a distilled (approximate) version. But in this, only half the number of parameters are used [23]. Specifically, token-type embeddings or a pooler are not included and only retained half layers from Google's BERT. Full output distributions can be attained after training an extensive neural network using a smaller network, which is the basic idea behind all this. Therefore, like a posterior approximation, it gives some sense. KulbackLeiber divergence of Bayesian Statistics is used as an optimization function for posterior approximation. The resultant confusion matrix of DISTILL-BERT is shown in Figure 14, training and validation accuracies are mentioned in Table 1, and the complete performance metrics are updated in Table 2.

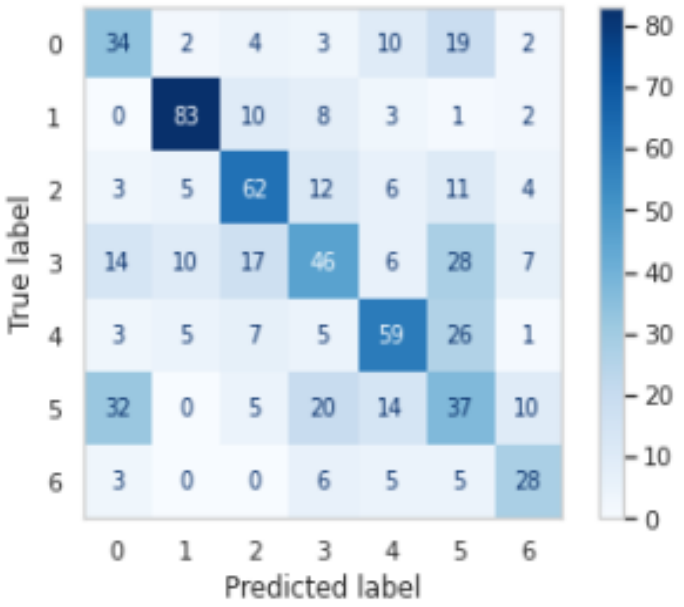



**Figure 14.** Confusion matrix of DISTILL-BERT

### 3.2.5. RoBERTa

RoBERTa is a more advanced version of BERT, SOTA proves it is 2.2% more efficient than BERT. Except for this, from BERT's pre-training, the Next Sentence Prediction (NSP) task is removed by RoBERTa. Further, during the training, the masked token is changed by dynamic masking. The benefits of large batch sizes during the training procedure are also impactful. The resultant confusion matrix of RoBERTa is shown in Figure 15, training and validation accuracies are mentioned in Table 1, and the complete performance metrics are updated in Table 2.

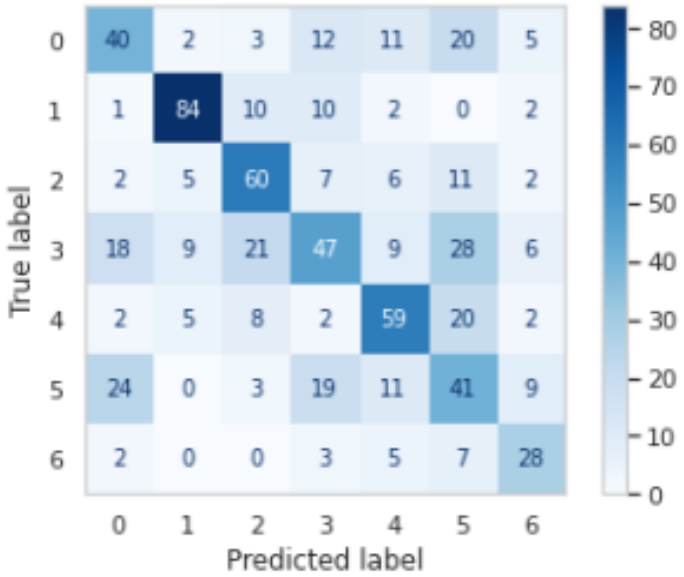



**Figure 15.** Confusion matrix of RO-BERT

### 3.2.6 XL-NET

XLNet attained impressive prediction metrics on the task of 20 languages compared to BERT, as it uses extensive computational power, huge data, and a large bidirectional transformer. It introduces permutation language modeling by randomly predicting all the tokens to enhance the training. Further, compared to BERT's masked language model or traditional language model where 15% masked tokens or sequentially tokens are generated respectively, XL-Net generates and predicts all the tokens randomly. Therefore, the model can learn the bidirectional relationships, and moreover, it can handle dependencies and relations better between words. In addition, the model has shown an excellent result even in the absence of permutation-based training, as it uses Transformer XL as the base architecture. The resultant confusion matrix of XL-Net is shown in Figure 16, training

and validation accuracies are mentioned in Table 1, and the complete performance metrics are updated in Table 2.

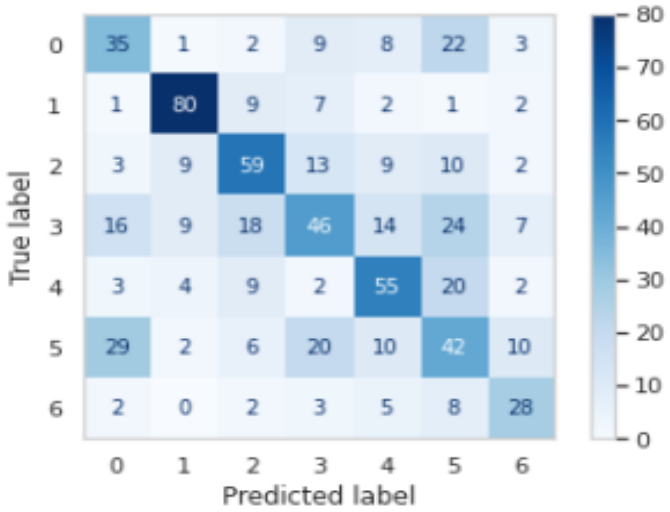



**Figure 16.** Confusion matrix of XL-NET

### 3.2.7 CuDNN-LSTM WITH 5-FOLD ATTENTION

CuDNN-LSTM is a modified version of the LSTM engine specifically designed for GPU (Compute Unified Device Architecture (CUDA) parameters). We haven't used simple LSTM we were computing NVIDIA GPU apart from this benefit of using this is GPUs are good for massively parallel computation, most of the linear algebra ops can be parallelized to improve performance, and Vector operations like matrix multiplication and gradient descent can be applied to large matrices that are executed in parallel with GPU support. CUDA provides an interface that allows vector ops to take advantage of GPU parallelism. CuDNN implements kernels for large matrix operations on GPU using CUDA. Here, CuDNNLSTM is designed for CUDA parallel processing and cannot run without GPU. But LSTM is designed for normal CPUs. The faster time of execution is because of parallelism. The resultant confusion matrix of CuDNN LSTM with 5-FOLD attention is shown in Figure 17, training and validation accuracies are mentioned in Table 1, and the complete performance metrics are updated in Table 2.

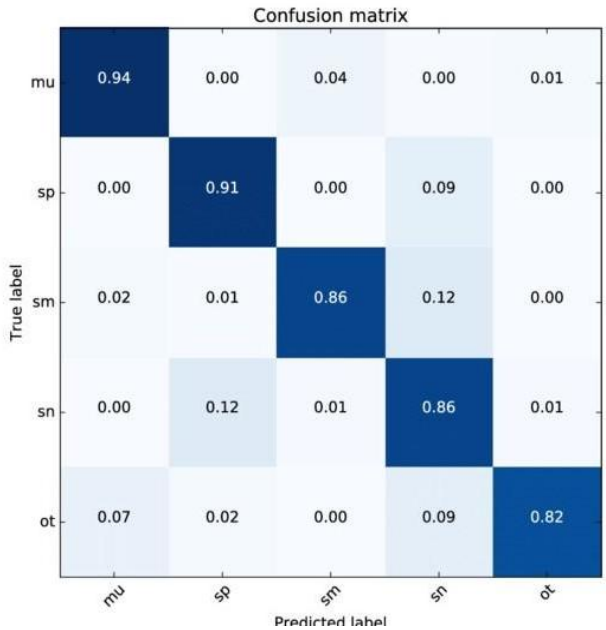



**Figure 17.** Confusion matrix of CuDNN LSTM with 5-FOLD attention

Training, testing, and validation accuracy of Image Captioning and Sentiment Analysis models are discussed in Table 1.

**Table 1.** Accuracies analysis of various models

| **Image Captioning Model** | **Training Accuracy** | **Testing Accuracy** | **validation accuracy on an unseen dataset** | **Sentiment Analysis Model** | **Training Accuracy** | **Testing Accuracy** | **validation accuracy on an unseen dataset** |
|---|---|---|---|---|---|---|---|
| Mobile Net LSTM | 81.97 % | 72.34 % | 80 % | BERT | 71.9 % | 50.92 % | 34 % |
| Inceptionnet V2 LSTM | 89.7 % | 71.2 % | 70.7 % | Ro-Bert | 69% | 51 % | 29.1 % |
| Noisy Student Efficient Net | 60.8 % | 52.3 % | 42.4 % | DistilBert | 64 % | 55 % | 30 % |
| VGG16 Attention | 88.56 % | 82.18 % | 56.4 % | XLNET | 52.4 % | 50.9 % | 23.9 % |
| Show Attend & Tell | 87.92 % | 82.3 % | 75 % | LSTM word to vector | 55 % | 30 % | 72.8 % |
| **VGG 16 LSTM** | **94.5 %** | **92.9 %** | **87.9 %** | **CuDNN LSTM with 5-fold Attention** | **92 %** | **82 %** | **81.1 %** |

Precision, Recall, F1 score, and sensitivity for the sentiment analyzers are mentioned in Table 2.

**Table 2.** Performance metrics analysis of various models

| **Model Name** | **Sensitivity** | **Precision** | **Recall** | **F1- Score** |
|---|---|---|---|---|
| BERT | 0.50 | 0.51 | 0.52 | 0.41 |
| Ro-Bert | 0.51 | 0.53 | 0.54 | 0.50 |
| DistilBert | 0.52 | 0.52 | 0.52 | 0.51 |
| XLNET | 0.49 | 0.50 | 0.50 | 0.51 |
| LSTM word to vector | 0.49 | 0.51 | 0.52 | 0.51 |
| CuDNN LSTM with 5-fold Attention | 0.54 | 0.60 | 0.58 | 0.54 |

## 4. Conclusion and Future Scope

Mood Prediction using Landscapes is an abstract problem that discusses multiple aspects like Colour Psychology, Object context, object-to-object relationships, and many more. Experiments with various possible architectures and customized algorithms for color psychology and object-to-context relationships are carried out. The solution involves context and concept-aware caption generation of Landscape images and analyzing the sentiment of the caption to predict the mood. Good accuracies are observed with VGG 16 LSTM as an Image captioning technique and CuDNN LSTM with 5 Folds of attention as sentiment analysis of 94 % of training and 87 % of real-time production accuracy on an unseen dataset. This work can be further extended to make the approach un-supervised and deploy object-to-object relationship factors to improve model judgments. Future works will involve moving this algorithm from image to real-time video and generating an automated video filtering technique based on landscape moods.

**Data Availability:** To do the implementation, the dataset is considered from three different sources mentioned in the reference section as reference numbers 12, 13, and 14.

**Funding Statement:** The author(s) received no funding for this study.

**Conflicts of Interest:** The authors declare they have no conflicts of interest to report regarding the present study.

**References**

[1] D. R. Williams and M. E. Patterson, "Environmental psychology: Mapping landscape meanings for ecosystem management," *In: Cordell, H. K.; Bergstrom, J. C., eds. Integrating social sciences and ecosystem management: Human dimensions in assessment, policy and management. Champaign, IL: Sagamore Press. p. 141-160.*, 1999, Accessed: Jan. 23, 2026. [Online]. Available: https://research.fs.usda.gov/treesearch/29423

[2] R. S. Ulrich, "Visual Landscapes and Psychological Weil-Being," *Landsc. Res.*, vol. 4, no. 1, pp. 17–23, Mar. 1979, doi: 10.1080/01426397908705892;Journal:Journal:Clar20;Requestedjournal:Journal:Clar20;Wgroup:String:Publication.

[3] B. Kara, "Landscape Design and Cognitive Psychology," *Procedia Soc. Behav. Sci.*, vol. 82, pp. 288–291, Jul. 2013, doi: 10.1016/J.SBSPRO.2013.06.262.

[4] T. Mittal, U. Bhattacharya, R. Chandra, A. Bera, and D. Manocha, "M3ER: Multiplicative Multimodal Emotion Recognition using Facial, Textual, and Speech Cues," *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 34, no. 02, pp. 1359–1367, Apr. 2020, doi: 10.1609/AAAI.V34I02.5492.

[5] A. Jan, H. Meng, Y. F. A. Gaus, F. Zhang, and S. Turabzadeh, "Automatic depression scale prediction using facial expression dynamics and regression," *AVEC 2014 - Proceedings of the 4th International Workshop on Audio/Visual Emotion Challenge, Workshop of MM 2014*, pp. 73–80, Nov. 2014, doi: 10.1145/2661806.2661812;Topic:Topic:Conference-Collections.

[6] "Using automated facial expression analysis for emotion and behavior prediction." Accessed: Jan. 23, 2026. [Online]. Available: https://psycnet.apa.org/record/2013-41550-020

[7] Z. H. Kilimci, A. Güven, M. Uysal, and S. Akyokus, "Mood Detection from Physical and Neurophysical Data Using Deep Learning Models," *Complexity*, vol. 2019, no. 1, p. 6434578, Jan. 2019, doi: 10.1155/2019/6434578.

[8] "Emotional Impact of Color in Landscape Photography." Accessed: Jan. 23, 2026. [Online]. Available: https://visualwilderness.com/composition-creativity/emotional-impact-of-color-in-landscape-photography

[9] T. Rao, X. Li, and M. Xu, "Learning Multi-level Deep Representations for Image Emotion Classification," *Neural Process. Lett.*, vol. 51, no. 3, pp. 2043–2061, Jun. 2020, doi: 10.1007/S11063-019-10033-9;Requestedjournal:Journal:Nple;Page:String:Article/Chapter.

[10] S. H. Wang, M. A. Khan, and Y. D. Zhang, "VISPNN: VGG-inspired Stochastic Pooling Neural Network," *Computers, materials & continua*, vol. 70, no. 2, pp. 3081–3097, 2022, doi: 10.32604/CMC.2022.019447.

[11] S. H. Wang, S. L. Fernandes, Z. Zhu, and Y. D. Zhang, "AVNC: Attention-Based VGG-Style Network for COVID-19 Diagnosis by CBAM," *IEEE Sens. J.*, vol. 22, no. 18, pp. 17431–17438, Sep. 2021, doi: 10.1109/JSEN.2021.3062442.

[12] "Landscape Pictures." Accessed: Jan. 23, 2026. [Online]. Available: https://www.kaggle.com/datasets/arnaud58/landscape-pictures

[13] J. Brejcha and M. Cadik, "GeoPose3K: Mountain Landscape Dataset for Camera Pose Estimation in Outdoor Environments," *Image Vis. Comput.*, Jun. 2017.

[14] "1500+ Nature Landscape Pictures | Download Free Images on Unsplash." Accessed: Jan. 23, 2026. [Online]. Available: https://unsplash.com/s/photos/nature-landscape

[15] "CLIP." Accessed: Jan. 23, 2026. [Online]. Available: https://huggingface.co/docs/transformers/model_doc/clip

[16] M. Chen *et al.*, "Generative Pretraining From Pixels," Nov. 21, 2020, *PMLR*. Accessed: Jan. 23, 2026. [Online]. Available: https://proceedings.mlr.press/v119/chen20s.html

[17] H. Akbari *et al.*, "VATT: Transformers for Multimodal Self-Supervised Learning from Raw Video, Audio and Text," *Adv. Neural Inf. Process. Syst.*, vol. 29, pp. 24206–24221, Apr. 2021, Accessed: Jan. 23, 2026. [Online]. Available: https://arxiv.org/pdf/2104.11178

[18] C. J. Buys, "Freud in Introductory Psychology Texts," *Teaching of Psychology*, vol. 3, no. 4, pp. 160–167, 1976, doi: 10.1207/S15328023TOP0304_2.

[19] R. Tarafdar, "Algorithms on Majority Problem," 2017. Accessed: Jan. 23, 2026. [Online]. Available: https://hdl.handle.net/10355/62441

[20] L. Luo, H. Yang, and F. Y. L. Chin, "EmotionX-DLC: Self-Attentive BiLSTM for Detecting Sequential Emotions in Dialogue," *Proceedings of the Annual Meeting of the Association for Computational Linguistics*, pp. 32–36, Jun. 2018, doi: 10.18653/v1/w18-3506.

[21] R. K. Yadav, L. Jiao, M. Goodwin, and O. C. Granmo, "Positionless aspect based sentiment analysis using attention mechanism," *Knowl. Based. Syst.*, vol. 226, p. 107136, Aug. 2021, doi: 10.1016/J.KNOSYS.2021.107136.

[22] B. Dickinson, M. Ganger, W. Hu, B. Dickinson, M. Ganger, and W. Hu, "Dimensionality Reduction of Distributed Vector Word Representations and Emoticon Stemming for Sentiment Analysis," *Journal of Data Analysis and Information Processing*, vol. 3, no. 4, pp. 153–162, Nov. 2015, doi: 10.4236/JDAIP.2015.34015.

[23] V. Sanh, L. Debut, J. Chaumond, and T. Wolf, "DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter," Oct. 2019, Accessed: Jan. 23, 2026. [Online]. Available: https://arxiv.org/pdf/1910.01108