الجمهورية الجزائرية الديمقراطية الشعبية
وزارة التعليم العالي و البحث العلمي

Université Abou Bekr Belkaid
Tlemcen Algérie
جامعة أبي بكر بلقايد
تلمسان الجزائر

PEOPLE'S DEMOCRATIC REPUBLIC OF ALGERIA
Ministry of Higher Education And Scientific Research

University of Tlemcen – ABOU BEKR BELKAID - Algeria
FACULTY OF SCIENCE
DEPARTMENT of COMPUTER SCIENCE

Graduation Project

To obtain a Master's degree in Computer Science

Specialty: Intelligent Model and Decision (M.I.D)

On the subject :

Multimodal Deep Learning for Visual Question Answering: Techniques and Challenges

Presented by:
Mr. MALFI Oualid Ilyes
Supervised by:
Mr. HADJILA Fethallah
Examined by:
Mr. CHOUITI Sidi Mohammed (President)
Mr. BENAZZOUZ Mourtada (Examiner)

Academic year: 2023/2024

Acknowledgments

First, I thank God the Almighty for giving me the strength, courage, and patience to complete this modest work.

I extend my heartfelt gratitude to my supervisor, Mr. Hadjila Fethallah. I am deeply thankful for your guidance, unwavering support, and constructive critiques, which contributed immensely to the successful completion of this work.

I also thank the jury members who agreed to read and evaluate this work. Your time and expertise are greatly appreciated.

I am sincerely grateful to all the individuals who have directly or indirectly contributed to this work. Your contributions, whether big or small, have played a significant role in shaping my academic journey and the successful completion of this research.

Dedication

This thesis is dedicated to my family, whose love and support have been my source of strength and inspiration. To my parents, for their endless sacrifices and encouragement. To my friends and colleagues, for their unwavering support and motivation. And to my mentor, Mr. Hadjila Fethallah, for his invaluable guidance and dedication.

Thank you all for believing in me and making this journey possible.

Contents

List of Figure

List of Abbreviations

VQA	Visual Question Answering
CV	Computer vision
NLP	Natural Language Processing
AI	Artificial intelligence
ML	Machine learning
DL	Deep learning
CNN	Convolutional Neural Network
BERT	Bidirectional Encoder Representations from Transformers
LSTM	Long Short-Term Memory
MLM	Masked Language Model
VGG	Visual Geometry Group
ResNet	Residual Network
ViT	Vision Transformer
CLEVR	Compositional Language and Elementary Visual Reasoning
VQA-RAD	Visual Question Answering in Radiology
ReLU	Rectified Linear Unit
ADAM	Adaptive Moment Estimation

Introduction

Artificial Intelligence (AI) is a rapidly evolving field that strives to create machines capable of performing tasks that typically require human intelligence(Russell & Norvig, 2016). From recognizing speech and understanding language to making decisions and perceiving the visual world, AI aims to replicate and even surpass human abilities in various domains. This ambitious endeavor has led to significant advancements and innovations, fundamentally transforming numerous industries and everyday life.

At the heart of many AI advancements is Machine Learning (ML), a subset of AI focused on the development of algorithms that allow computers to learn from and make predictions or decisions based on data. ML has fueled innovations across various fields, including healthcare, finance, and transportation, by enabling systems to identify patterns, make informed decisions, and improve over time through experience(Janiesch et al., 2021).

Within the realm of ML lies Deep Learning, a more specialized subset that uses neural networks with many layers to model complex patterns in large datasets. Initially, deep learning models were primarily unimodal, meaning they processed a single type of data, such as images or text. These unimodal models have achieved remarkable success in tasks like image classification, speech recognition, and natural language processing (Baltrušaitis et al., 2017; LeCun et al., 2015) .

However, the real world is inherently multimodal. We perceive and interact with it through multiple senses simultaneously, such as seeing and hearing. Recognizing this, researchers have increasingly focused on developing multimodal deep learning models that can integrate and process information from multiple sources. These models aim to create a more holistic understanding by combining different modalities, such as visual, auditory, and textual data(Baltrušaitis et al., 2017).

1.1. Problem Statement

One of the most exciting applications of multimodal deep learning is Visual Question Answering (VQA)(Wu et al., 2016). VQA systems are designed to answer questions about visual content, combining insights from computer vision and natural language processing. This capability is particularly powerful, as it allows machines to interpret and reason about complex scenes in a manner similar to humans (Baltrušaitis et al., 2017).

1.2. Motivation

Selecting a VQA project for my master's study is both motivating and challenging, covering the vibrant fields of computer vision, NLP, and AI. VQA integrates data from these rapidly growing areas and demonstrates how machines can see and understand visual imagery. VQA has applications in various fields, such as autonomous vehicles, medical image analysis, assistive technology, and content-based image search. These systems can answer questions about image content in a human-like manner, presenting numerous technical challenges, including scene comprehension and data fusion across different modalities.

Engaging in VQA research will expose me to the latest advancements in deep learning architectures and multimodal fusion techniques, allowing me to contribute to cutting-edge AI research. This project will also provide practical experience in using state-of-the-art deep learning frameworks, enhancing my critical thinking and problem-solving skills. Moreover, VQA projects offer flexibility in research direction, allowing me to focus on areas of personal interest within computer vision or natural language processing.

VQA has significant societal impact potential, improving accessibility and aiding visually impaired individuals, making the work deeply fulfilling. Collaborating with researchers from diverse backgrounds and establishing connections within academic and industry communities are additional benefits. Lastly, expertise in VQA and related fields is highly sought after, positioning me for exciting career opportunities in research, academia, and industry. Thus, embarking on a VQA project promises to expand my technical skills and foster personal and professional growth as I contribute to advancing the frontiers of artificial intelligence.

1.3. Contribution

In this study, we developed a Visual Question Answering (VQA) model that processes both textual and visual inputs (RGB images). For the visual component, we employed the VGG19 model for image feature extraction, and for the textual component, we used a BERT-based model to extract question features. Initially, each modality extracts features independently. We then combined these features using a mid-level fusion method through element-wise multiplication. This combined representation was fed into a VQA model composed of a fully connected layer, which maps the combined features to the output labels.

1.4. Outline of the manuscript

Chapter 2 provides an overview of machine learning and delves into deep learning in both computer vision and natural language processing.

Chapter 3 presents multimodal deep learning, including the five challenges.

Chapter 4 explains the VQA system proposed and the result, including all details.

Finally, we conclude by summarizing and talking about future work.

1. Deep Learning

1.1. Introduction to Machine Learning

Machine learning is a subset of artificial intelligence (AI) introduced by Arthur Samuel in 1959 that enables systems to learn and improve from experience without being explicitly programmed. It can be broadly categorized into three types (Figure 2.1): supervised learning, unsupervised learning, and reinforcement learning(Janiesch et al., 2021).

1.1.1. Supervised Learning

In supervised learning, the model is trained on a labeled dataset, which means that each training example is paired with an output label. The goal is to learn a mapping from inputs to outputs(Cunningham et al., 2008), which can be used to predict the output for new inputs. This approach has numerous applications, including image classification(Lu & Weng, 2007) ,spam detection (Crawford et al., 2015), and regression analysis(Sykes, 1993).To achieve this goal, several key algorithms have been developed Linear regression, support vector machines (Yue et al., 2003), and neural networks.

1.1.2. Unsupervised Learning

Unsupervised learning involves training a model on data without labeled responses. The model tries to identify patterns and structures in the input data(Janiesch et al., 2021).This approach finds applications in various domains, such as clustering , Dimensionality reduction techniques (Sorzano et al., 2014), Central to achieving this objective are a set of key algorithms K-means clustering, hierarchical clustering and principal component analysis (PCA).

1.1.3. Reinforcement Learning

In reinforcement learning, an agent interacts with an environment and learns to perform actions to maximize cumulative reward. It involves learning a policy that maps states of the environment to actions. Reinforcement learning has driven advancements in various fields, including robotics, game playing and autonomous driving. The field leverages a repertoire of key algorithms, including Q-learning, deep Q-networks(Arulkumaran et al., 2017).

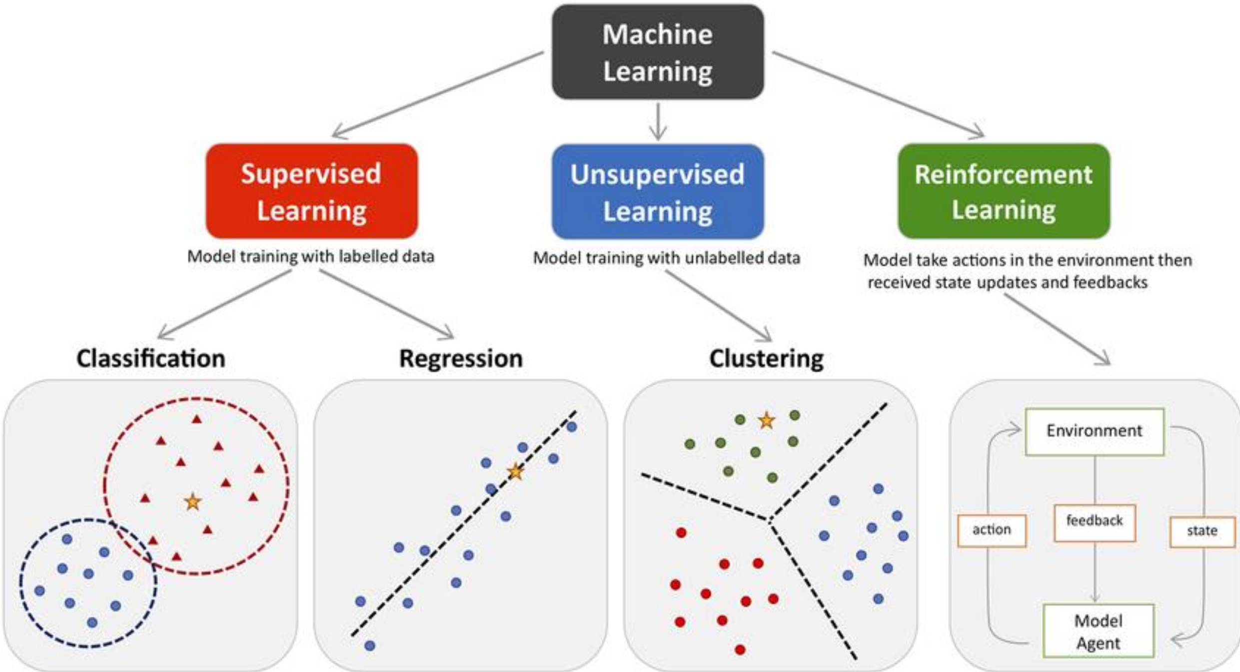


Figure 2. 1 The main types of machine learning (The Main Types of Machine Learning. Main Approaches Include. . ., s. d.)

1.2. Fundamentals of Deep Learning

Deep learning, introduced by Geoffrey Hinton, Yoshua Bengio, and Yann LeCun, is a subfield of machine learning that focuses on neural networks with many layers, capable of learning complex patterns in large amounts of data(LeCun et al., 2015). This approach has led to significant advancements in various fields by enabling the modeling of intricate data representations. The training process of deep learning models involves several key concepts and steps.

1.2.1. Neural Networks

A. Artificial Neurons

Artificial neurons are the basic units of a neural network, mimicking the functionality of biological neurons. Each neuron receives input, processes it, and passes the output to the next layer(McCulloch & Pitts, 1943)

B. Layers

Neural networks are organized into layers. The input layer receives the raw data, the hidden layers process the data through a series of transformations, and the output layer produces the final prediction.

C. Activation Functions

Activation functions introduce non-linearity into the network, enabling it to learn complex patterns. Common activation functions include ReLU (Rectified Linear Unit), sigmoid, and tanh (hyperbolic tangent)(Sharma et al., 2017)..

ReLU: Defined as $f(x) = max(0, x)$ It is widely used due to its simplicity and effectiveness in mitigating the vanishing gradient problem.

Sigmoid: Defined as $f(x) = \frac{1}{1+e^{-x}}$ it maps inputs to a range between 0 and 1.

Tanh: Defined as $f(x) = tanh(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}}$, it maps inputs to a range between -1 and 1.

As shown in Figure 2.2, these activation functions have distinct curves that illustrate their different mappings.

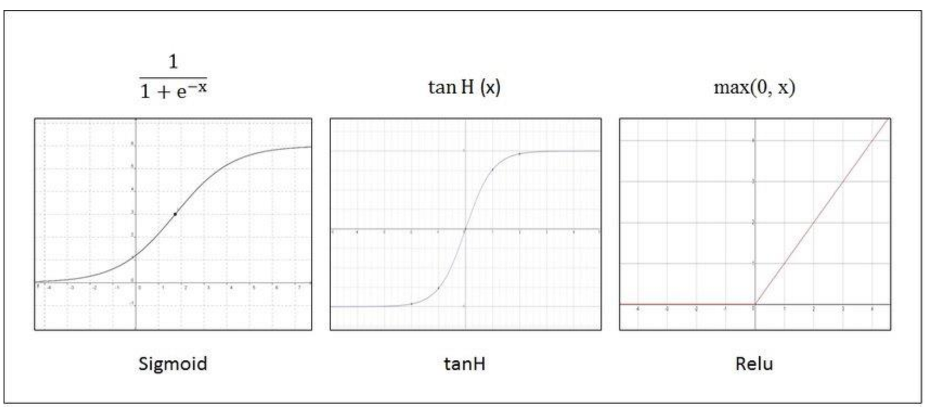


Figure 2. 2 Activation functions curves (This Figure Shows Three Different Activation Function's. . ., s. d.)

1.2.2. Training Process

A. Forward Propagation

During forward propagation, input data passes through the network layer by layer to generate predictions. Each layer transforms the input data using weights, biases, and activation functions.

B. Loss Function

The loss function measures the discrepancy between the predicted values and the actual target values. It provides a quantitative assessment of how well the model is performing. Common loss functions include mean squared error (MSE) for regression tasks and cross-entropy loss for classification tasks.

C. Backpropagation

Backpropagation is the process of adjusting the weights and biases of the network to minimize the loss function. It involves computing the gradient of the loss function with respect to each weight by applying the chain rule. These gradients are then used to update the weights in the direction that reduces the loss, typically using optimization algorithms such as stochastic gradient descent (SGD) or Adam

1.3. Deep learning in computer vision

In VQA, Computer Vision (CV) plays a crucial role in extracting and interpreting visual features from images. Techniques such as Convolutional Neural Networks (CNNs) are employed to learn hierarchical representations of the visual data, enabling the system to recognize objects, scenes, and other relevant visual cues (Alzubaidi et al., 2021). These visual features are essential for accurately understanding the context of the image in relation to the posed question(Wu et al., 2016) .

1.3.1. Convolutional Neural Network

A Convolutional Neural Network (CNN) is a type of deep learning model primarily used for processing and analyzing visual data. It is designed to automatically and adaptively learn spatial hierarchies of features through backpropagation by using multiple building blocks such as convolution layers, pooling layers, and fully connected layers (Alzubaidi et al., 2021)as illustrated in Figure 2.3.

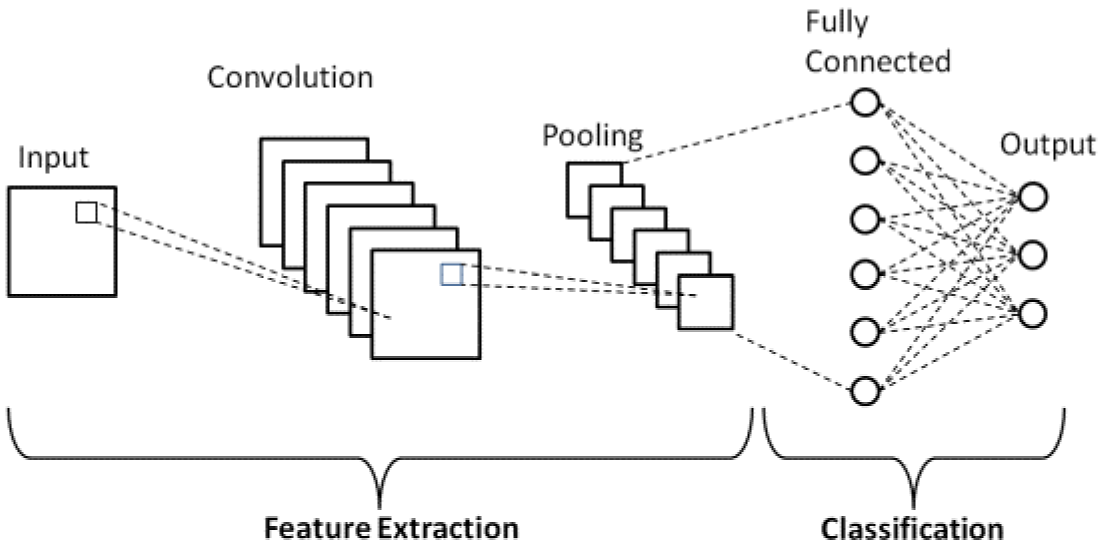


Figure 2. 3 CNN architecture [03]

A. Convolutional Layers

Convolutional layers are fundamental components of CNNs, designed to automatically and adaptively learn the spatial hierarchies of features from input images. These layers are composed of a series of filters, also known as kernels, that traverse the input data to generate feature maps, which are essential for identifying and extracting relevant features such as edges, textures, and patterns.Let's go through the details of its operation (Alzubaidi et al., 2021):

Input Image: The convolutional layer receives an input image represented as a matrix of pixel values.

Filter Application: A filter, which is a smaller matrix of weights, convolves across the input image. At each position, the filter performs an element-wise multiplication followed by a summation of the results to produce a single output value. This operation, repeated across the entire image, results in the creation of a feature map.

Activation Function: The resulting feature map is passed through an activation function, typically the Rectified Linear Unit (ReLU), to introduce non-linearity into the model. This step is crucial for the network's ability to learn complex patterns.

Output Feature Map: The final output of the convolutional layer is a new matrix, known as the feature map, which emphasizes the presence and locations of specific features detected by the filter.

This process is depicted in Figure 2.4.

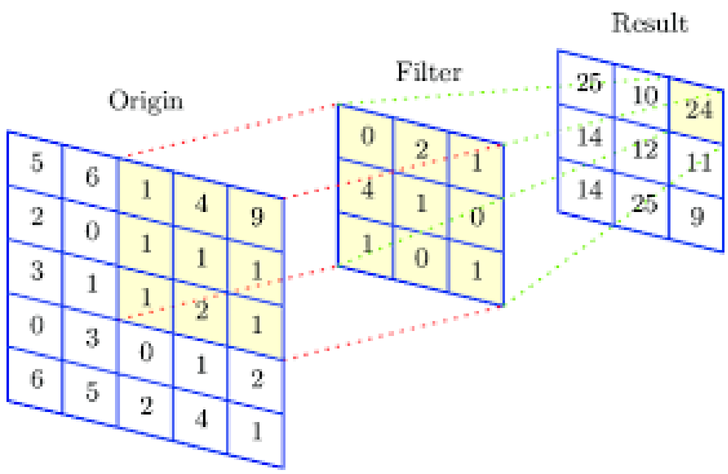


Figure 2. 4 convolutional operation example(Balaji, 2023)

B. Pooling Layers

Pooling layers are used to reduce the spatial dimensions of the feature maps, which helps in reducing the number of parameters and computation in the network. There are two common types of pooling: max pooling and average pooling, this is visualized in Figure 2.5.

Max pooling is an operation that extracts the most prominent features from the input feature map by selecting the maximum value within each patch of the feature map. This process helps in retaining the most significant features while reducing spatial dimensions. Here is an explanation of how it works.

- Patch Selection: The input feature map is divided into non-overlapping patches of a specified size, commonly 2x2.
- Max Value Selection: For each patch, the maximum value is identified and retained in the output feature map.
- Output Generation: The output is a downsampled version of the input feature map, where each retained value corresponds to the maximum value of the respective patch.

Average pooling, in contrast to max pooling, calculates the average value within each patch of the feature map. This operation is used to downsample the input while preserving the average presence of features across the patches. Here is an explanation of how it works .

- Patch Selection: Similar to max pooling, the input feature map is divided into non-overlapping patches of a specified size.

- Average Value Calculation: For each patch, the average value is computed.
- Output Generation: The output is a downsampled version of the input feature map, where each retained value is the average of the respective patch.

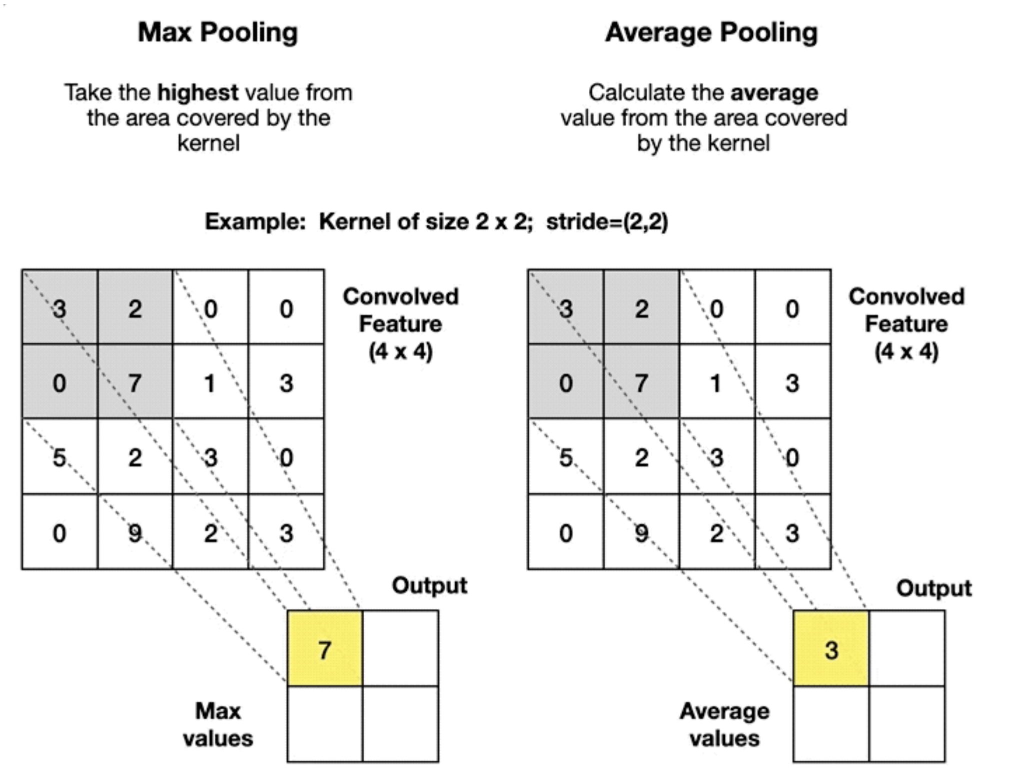


Figure 2. 5 example of pooling operation(Qayyum, 2022)

C. Fully Connected Layers

Fully connected layers, also referred to as dense layers, are typically utilized towards the end of a CNN. These layers play a crucial role in synthesizing the high-level features extracted by the preceding convolutional and pooling layers, thereby enabling the classification of the input image into distinct categories(see Figure 2.6).here is a description of its working process.

- Flattening: The feature maps produced by the final pooling layer are transformed into a one-dimensional vector. This process, known as flattening, converts the multidimensional feature maps into a format suitable for input into the fully connected layers.
- Matrix Multiplication: The flattened vector is then input into one or more fully connected layers. Each fully connected layer performs a matrix multiplication between the input vector and a weight matrix, followed by the addition of a bias vector. This operation aggregates the learned features and prepares them for classification.
- Activation Function: To introduce non-linearity into the model, the resulting values from the matrix multiplication are passed through an activation function. Common activation

functions used in fully connected layers include the Rectified Linear Unit (ReLU), which helps the network learn complex patterns by introducing non-linear transformations.

- Output Layer: The final fully connected layer, which is used for classification, typically employs a softmax activation function when dealing with multi-class classification problems. The softmax function converts the raw scores (logits) into probabilities, providing a probabilistic interpretation of the model's predictions.

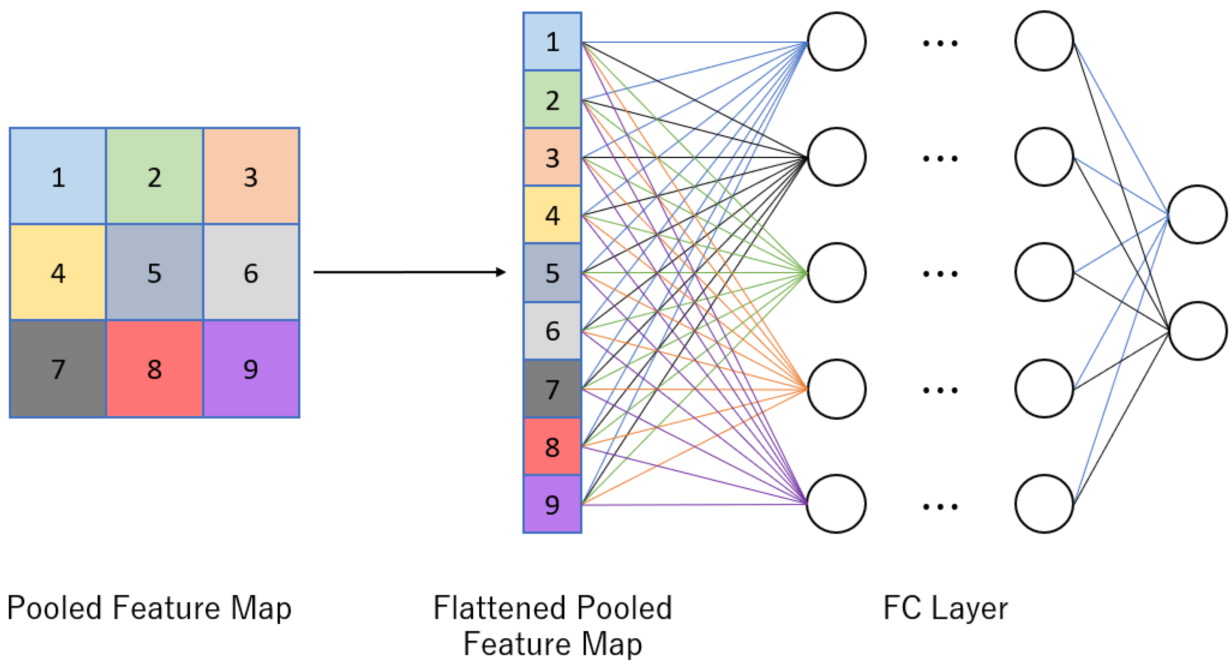


Figure 2. 6 A pooled feature map being flattened and inputted into a fully connected network layer(Aramendia, 2024)

1.3.2. CNN architectures

A. VGG Architecture

The Visual Geometry Group (VGG) network, designed by Karen Simonyan and Andrew Zisserman, is an influential model for large-scale image recognition. VGG networks utilize small 3x3 convolution filters and explore the performance of networks with varying depths, typically 16 to 19 layers. The VGG architecture emphasizes simplicity and uniformity, using only small-sized filters consistently across the network. The two primary versions, VGG16(The architecture can be seen in Figure 2.7) and VGG19, refer to models with 16 and 19 layers respectively(Vedaldi & Zisserman, 2016).

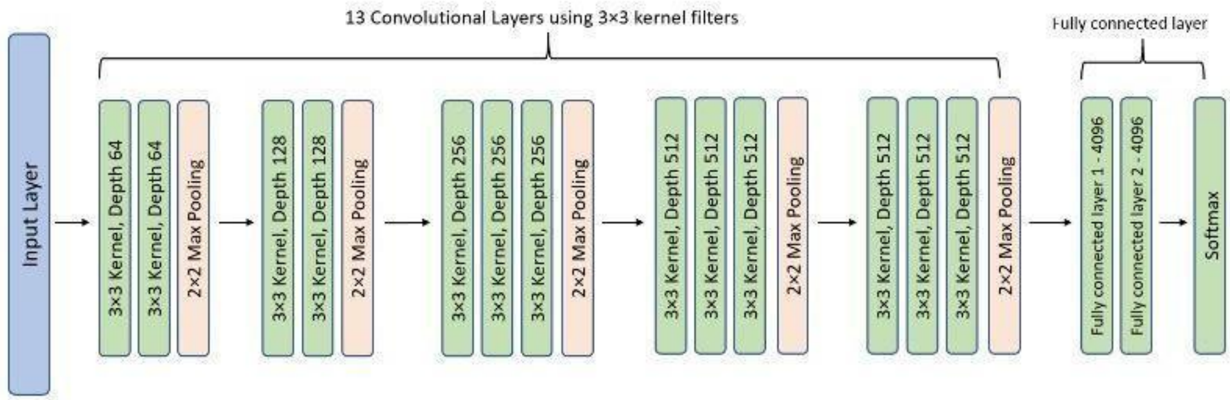


Figure 2. 7 VGG-16 network architectureBoesch (2024)

B. ResNet Architecture

Residual Networks (ResNet), introduced by He et al., address the problem of training very deep networks. By incorporating residual connections that allow gradients to flow through the network more effectively, ResNet models can be extremely deep—up to 152 layers (as in ResNet152)—without suffering from vanishing gradient issues. Figure 2.8 illustrates the architecture of a residual network, showcasing its distinctive residual connections that bypass one or more layers. the architecture's defining feature is its use of these shortcut connections, which enable the network to maintain performance as it scales in depth. This innovation ensures that deeper models like ResNet152 can learn effectively without degradation in training accuracy. ResNet's introduction of residual connections marked a significant advancement in deep learning model design.(Targ et al., 2016)

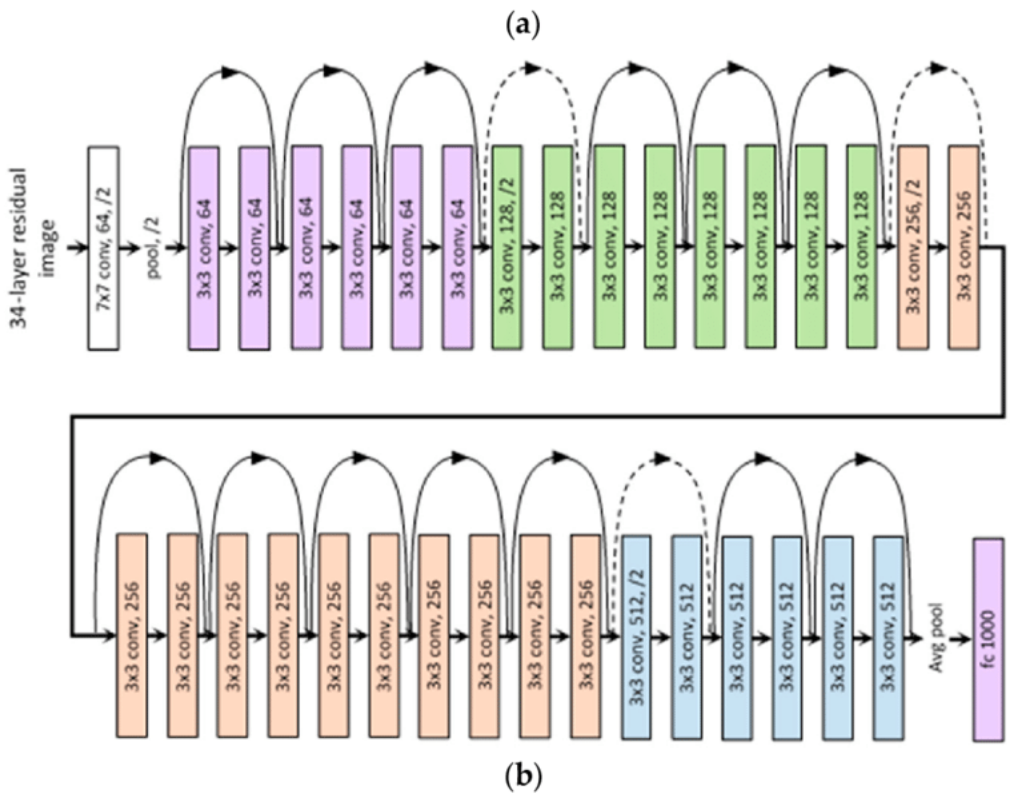


Figure 2. 8 residual network architecture (Bangar, 2022)

C. EfficientNet Architecture

EfficientNet, introduced by Mingxing Tan and Quoc V. Le from Google AI in 2019, employs a compound scaling method that uniformly scales network depth, width, and resolution, leading to more efficient and effective image recognition models with significantly lower computational and memory demands. The architecture balances these dimensions by using neural architecture search (NAS) to optimize the base model, EfficientNet-B0, which has approximately 82 layers. Subsequent models (EfficientNet-B1 to B7) progressively increase in layers, with EfficientNet-B1 around 88 layers and EfficientNet-B7 about 120 layers. This scaling method enhances the network's performance and efficiency, achieving state-of-the-art results across various benchmarks. For example, EfficientNet-B5 offers a strong balance between accuracy and efficiency, demonstrating significant performance gains over traditional models while maintaining reduced computational costs and memory usage.(Hoang & Jo, 2021; Tan & Le, 2019)

The various scaling methods used in EfficientNet can be seen in **Figure 1** below. This figure illustrates the baseline model and the effects of width scaling, depth scaling, resolution scaling, and compound scaling on the network architecture

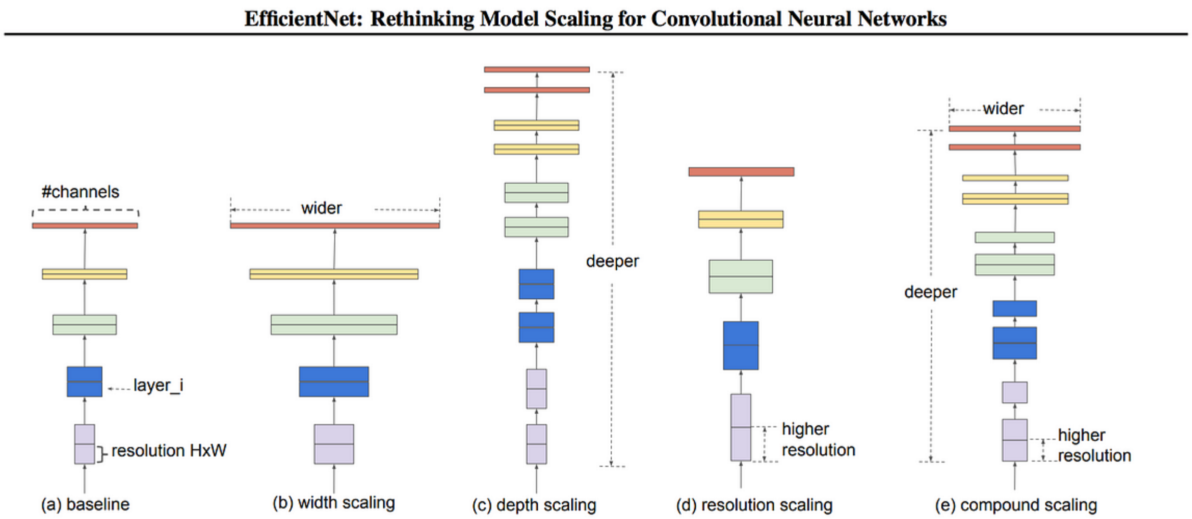


Figure 2. 9 EfficientNet architecture[19]

1.3.3. Vision Transformer (ViT)

A Vision Transformer (ViT) is a type of deep learning model that leverages the Transformer architecture, originally designed for natural language processing, to process and analyze visual data. Unlike Convolutional Neural Networks (CNNs), ViTs operate on image patches and utilize self-attention mechanisms to capture global relationships between different parts of an image (figure 2.10).(Dosovitskiy et al., 2021)

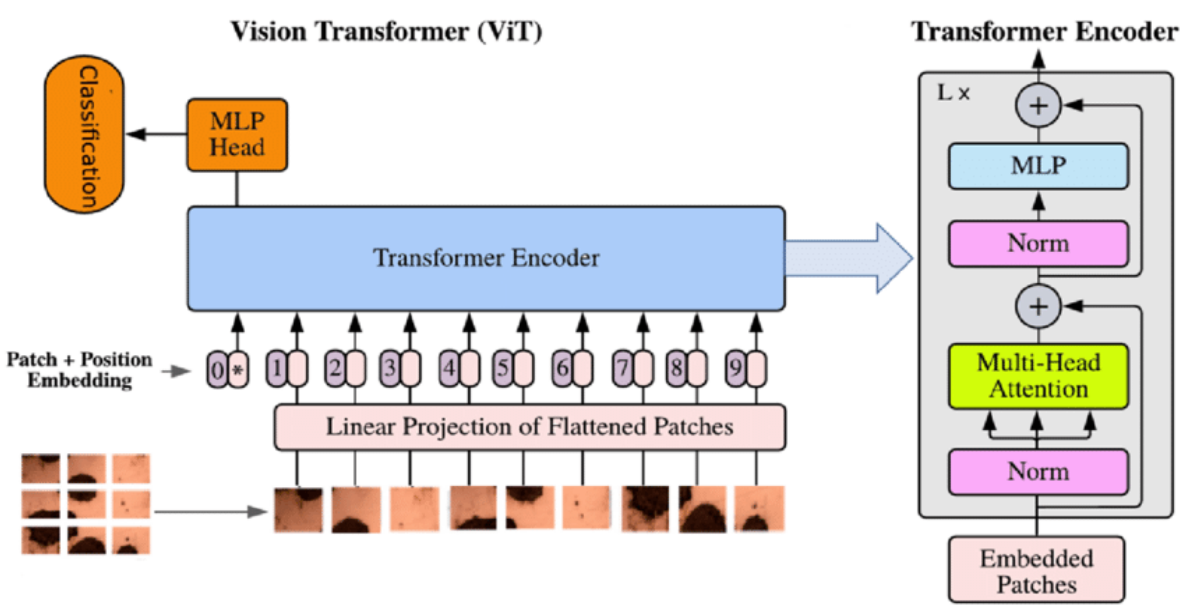


Figure 2. 10 Example of an architecture of the ViT (Vision Transformer (ViT) Model Architecture. ViT Breaks The. . ., s. d.)

A. Image Patch Embedding

Image Patch Embedding is the initial step in a Vision Transformer model, where the input image is divided into a sequence of smaller, non-overlapping patches. These patches are then linearly embedded into a lower-dimensional space, forming the input tokens for the Transformer model (Yue et al., 2003) . here is an explanation of the key stages involved in the process.

- Input Image: The model receives an input image, typically represented as a matrix of pixel values.
- Patch Division: The input image is divided into fixed-size patches, commonly 16x16 pixels. Each patch is flattened into a one-dimensional vector.
- Linear Embedding: Each flattened patch vector is passed through a linear layer (fully connected layer) to create a low-dimensional embedding. These embeddings represent the image patches as input tokens for the Transformer.
- Positional Encoding: Since Transformers do not inherently capture positional information, positional encodings are added to the patch embeddings to retain the spatial structure of the image. This step helps the model understand the relative positions of patches within the image.

B. Transformer Encoder

The Transformer Encoder is the core component of the Vision Transformer, responsible for learning the global relationships between the image patches through self-attention mechanisms. It consists of multiple layers, each containing a multi-head self-attention mechanism and a feed-forward neural network(Zhou et al., 2021).Below is an outline of the critical steps of the process.

- Multi-Head Self-Attention: The self-attention mechanism computes the relationships between all patches by producing attention scores. These scores indicate the importance of each patch relative to others. The multi-head approach allows the model to focus on different aspects of the patches simultaneously.
- Feed-Forward Network: After the self-attention mechanism, the output is passed through a feed-forward neural network, which consists of two linear layers with a non-linear activation function (typically GELU) in between.

- Residual Connections and Layer Normalization: Each attention and feed-forward block is followed by residual connections and layer normalization to stabilize training and enhance the model's performance.

C. Classification Head

The Classification Head is the final component of the Vision Transformer, responsible for synthesizing the high-level features extracted by the Transformer Encoder and producing the final classification output. the next part provides how it works:

- Class Token: A special learnable class token is prepended to the sequence of patch embeddings before they are input into the Transformer Encoder. This token aggregates the global information from all patches during the self-attention process.
- Aggregation: After passing through the Transformer Encoder, the final representation of the class token is used as the aggregated feature vector for the entire image.
- Fully Connected Layers: The aggregated feature vector is passed through one or more fully connected layers to perform classification. These layers are responsible for mapping the learned features to the target classes.
- Softmax Activation: In multi-class classification tasks, the final fully connected layer employs a softmax activation function to convert the raw scores (logits) into probabilities, providing a probabilistic interpretation of the model's predictions.

1.4. Deep learning in language natural processing

In Natural Language Processing (NLP), deep learning plays a pivotal role in understanding and generating human language. Techniques such as Recurrent Neural Networks (RNNs)(Elman, 1990), Long Short-Term Memory (LSTM) networks (Smagulova & James, 2019), and Transformers (Vaswani et al., 2017)are employed to learn complex patterns and dependencies in textual data. These models enable systems to perform tasks such as language translation, sentiment analysis, and question answering by capturing semantic and syntactic nuances in the text. The ability to comprehend context and generate coherent language makes deep learning essential for advancing NLP applications (Patwardhan et al., 2023).

1.4.1. Long Short-Term Memory

Long Short-Term Memory (LSTM) networks are advanced architectures of Recurrent Neural Networks (RNNs), designed for sequential data processing. Unlike traditional feedforward neural networks, LSTMs have connections that form directed cycles, allowing information to persist across time steps. Specifically designed to handle long-range dependencies and mitigate the vanishing gradient problem, LSTMs introduce memory cells and gating mechanisms to control the flow of information(Smagulova & James, 2019).The architecture can be seen in Figure 2.11.

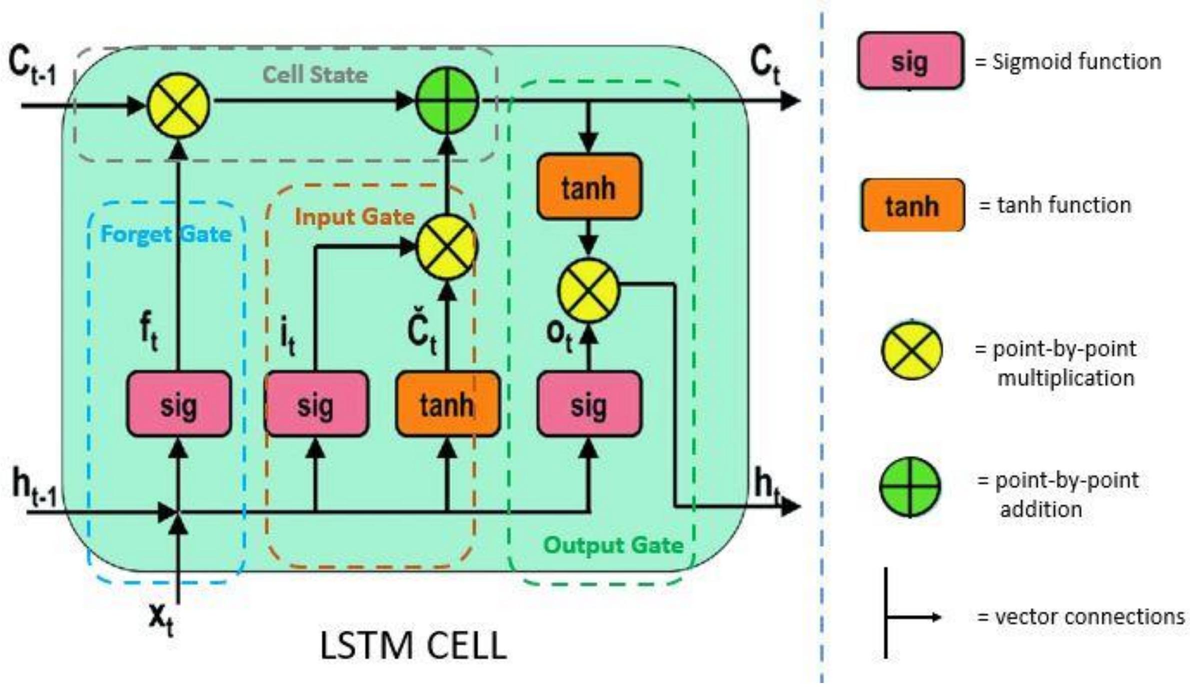


Figure 2. 11 Long Short-Term Memory networks (Singh ,2024)

Components of LSTM:

1. Cell State (Ct): A memory state that carries information across different time steps.

2. Gates: Three gates regulate the flow of information:

Forget Gate (ft): $ft = \sigma(Wf \cdot [ht - 1, xt] + bf)$

The forget gate determines which information from the previous cell state ($Ct - 1$)should be discarded. It uses a sigmoid activation function (σ) to produce an output between 0 and 1, where 0 represents completely forgetting the information and 1 represents completely retaining it.

Input Gate (it): $it = \sigma(Wi \cdot [ht - 1, xt] + bi)$

The input gate decides which new information will be added to the cell state (Ct). This gate also uses a sigmoid activation function to control the extent of the input that should be added. The combination of the input gate and a candidate cell state ($\check{C}t$) updates the cell state.

Output Gate (Ot): $ot = \sigma(Wo \cdot [ht - 1, xt] + bo)$

The output gate determines what part of the cell state should be output to the hidden state (ht). It uses a sigmoid function to control the amount of information from the cell state that influences the hidden state.

Cell State Update: $Ct = ft * Ct - 1 + it * \check{C}t$

where (Č) is the candidate cell state, typically calculated as: $\check{C}t = tanh(Wc \cdot [ht - 1, xt] + bc)$

The cell state (Ct) is updated by combining the old cell state ($Ct - 1$) and the candidate cell state ($\check{C}t$). The forget gate (ft) modulates the contribution of the previous cell state, while the input gate (it) and candidate cell state ($\check{C}t$) determine the new information to be added.

Hidden State Update: $ht = ot * tanh(Ct)$

The hidden state (ht) is updated by modulating the cell state (Ct) through the output gate (Ot) and applying a tanh activation function to it. The output gate decides which parts of the cell state should be passed to the next hidden state.

1.4.2. Bidirectional Encoder Representations from Transformers

BERT, introduced by Devlin et al. in 2018, is a transformative model in natural language processing (NLP) that excels at capturing context in text by considering both the left and right surroundings of a word simultaneously. BERT's architecture and training approach have significantly advanced the field of NLP, enabling more accurate language understanding and generation tasks(Acheampong et al., 2021; Kenton & Toutanova, 2019).

A. Architecture of BERT

BERT is built upon the Transformer architecture, specifically utilizing the encoder part of the Transformer. The Transformer model, introduced by Vaswani et al. in 2017, leverages self-attention mechanisms to process input text. BERT comes in two primary versions, distinguished by their size and complexity. BERT-Base consists of 12 layers (Transformer blocks as represented in Figure 2.12), 12 attention heads, and has a hidden size of 768, resulting in a total of 110 million parameters. On the other hand, BERT-Large is more complex, featuring 24 layers, 16 attention heads, and a hidden size of 1024, which collectively amount to 340 million parameters. These configurations allow BERT to capture intricate patterns in language data, making it highly effective for a wide range of natural language processing

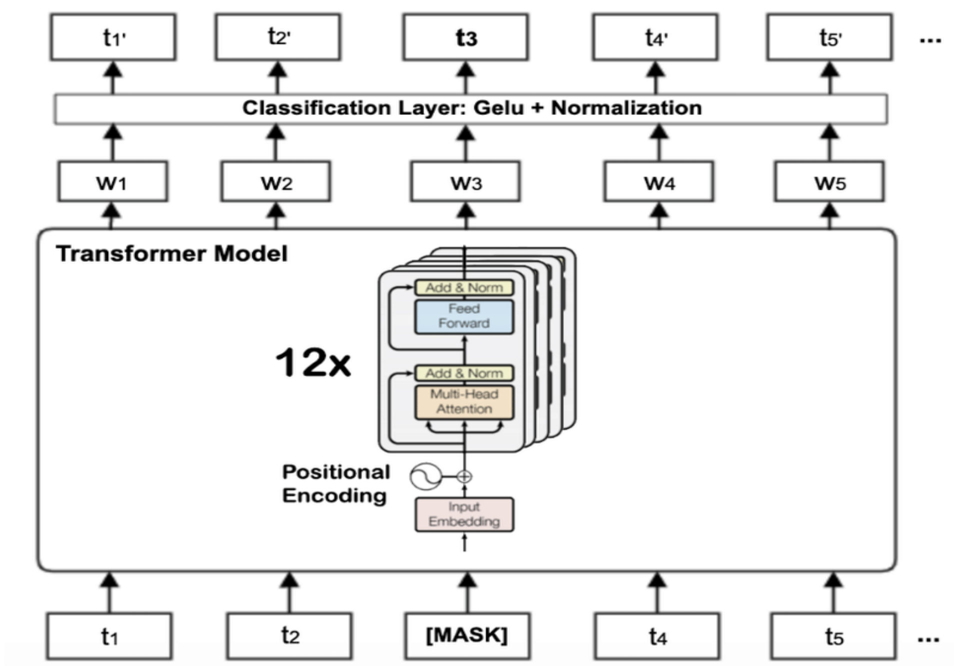


Figure 2. 12: The Transformer based BERT base architecture with twelve encoder blocks (The Transformer Based BERT Base Architecture with Twelve. . ., s. d.)

B. Bidirectional Context

Unlike previous models that processed text either left-to-right or right-to-left, BERT uses a bidirectional approach. This means it considers both previous and next words in context, allowing for a deeper understanding of word meaning and context.

C. Encoder Architecture

The encoder in BERT consists of several layers, each comprising two main sub-layers: self-attention and feedforward neural networks

a. **Self-Attention Mechanism**

- Query, Key, and Value Vectors: Each word in the input sequence is transformed into three vectors: Query (Q), Key (K), and Value (V). These vectors are obtained through linear transformations of the word embeddings.
- Attention Scores: The self-attention mechanism computes a score for each word pair by taking the dot product of the query vector of one word with the key vector of another. These scores are scaled by the square root of the dimension of the key vectors to ensure stable gradients. The scores are then passed through a SoftMax function to obtain the attention weights, which represent the importance of each word in the context of the current word.
- Contextual Representations: The attention weights are used to compute a weighted sum of the value vectors, producing a new representation for each word that incorporates contextual information from the entire sequence.

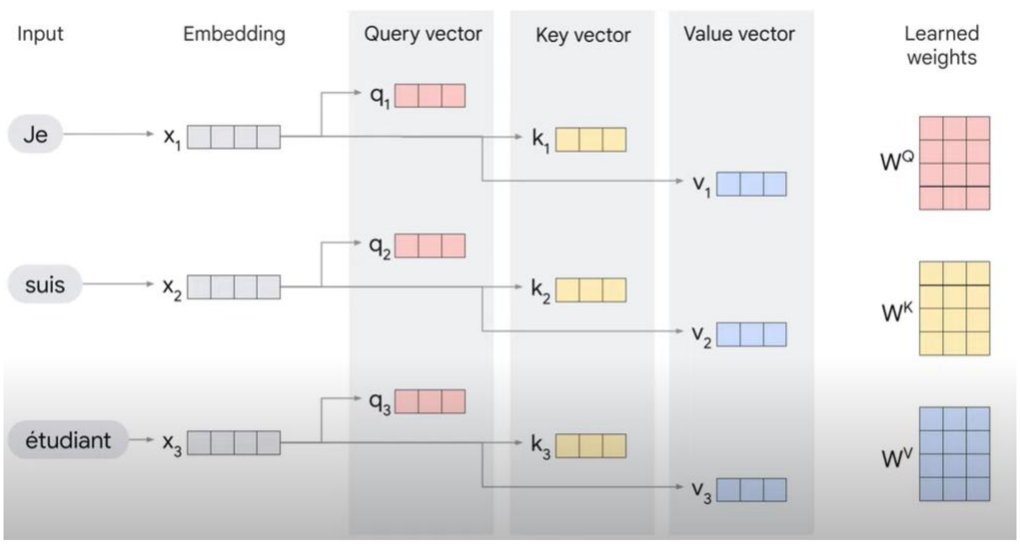


Figure 2. 13 Detailed Breakdown of Query, Key, and Value Vectors in Transformer Architecture

b. Feedforward Neural Networks

After the self-attention mechanism, the output for each word goes through a feedforward neural network:

This consists of two linear transformations with a ReLU activation in between.

The same feedforward network is applied independently to each position.

D. Training Mechanisms

BERT's training involves two unsupervised tasks to develop a strong understanding of language:

Masked Language Model (MLM): In MLM, 15% of the words in the input sequence are randomly masked, and the model is trained to predict these masked words based on the context provided by the surrounding words.

Next Sentence Prediction (NSP): NSP is designed to improve BERT's performance on tasks that require an understanding of the relationship between sentences. During training, the model receives pairs of sentences and is trained to predict whether the second sentence is the actual next sentence in the original text or a randomly chosen one.

E. Input Embeddings

BERT uses three types of embeddings to represent the input text:

- Token Embedding: Represents the individual words or subwords.
- Segment Embedding: Differentiates between sentences in tasks like NSP.
- Position Embedding: Encodes the position of each word in the sequence to capture the order of words.

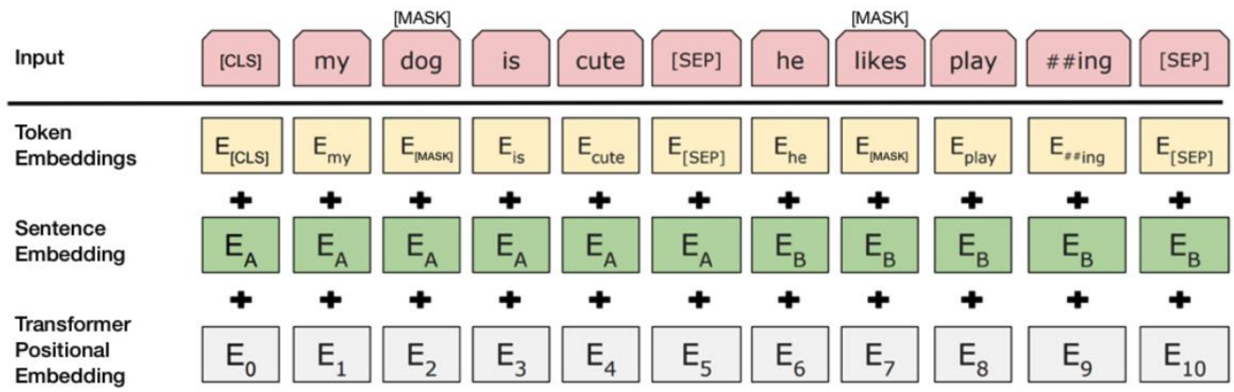


Figure 2. 14 BERT input representation. The input embeddings are the sum of the token embeddings, the segmentation embeddings and the position embeddings(Kenton & Toutanova, 2019)

2.5. Conclusion

In this chapter, we explored the foundations of machine learning and deep learning, emphasizing key concepts and architectures such as CNNs and Transformers. These models are crucial for advancements in computer vision and natural language processing, forming the basis for complex applications like Visual Question Answering (VQA), which requires the integration of visual and textual data to generate meaningful responses.

2. MultiModal deep Learning

2.1. Introduction

Multimodal deep learning is an evolving field that focuses on integrating information from multiple sensory modalities such as text, image, and audio. This integration aims to leverage the complementary nature of these modalities to enhance learning and inference. This chapter delves into the five primary challenges (as visualized in Figure 3.1) in multimodal deep learning: representation, translation, alignment, fusion, and co-learning. Each section will explore the fundamental concepts and subtleties within these challenges(Baltrušaitis et al., 2017; Jabeen et al., 2023) .

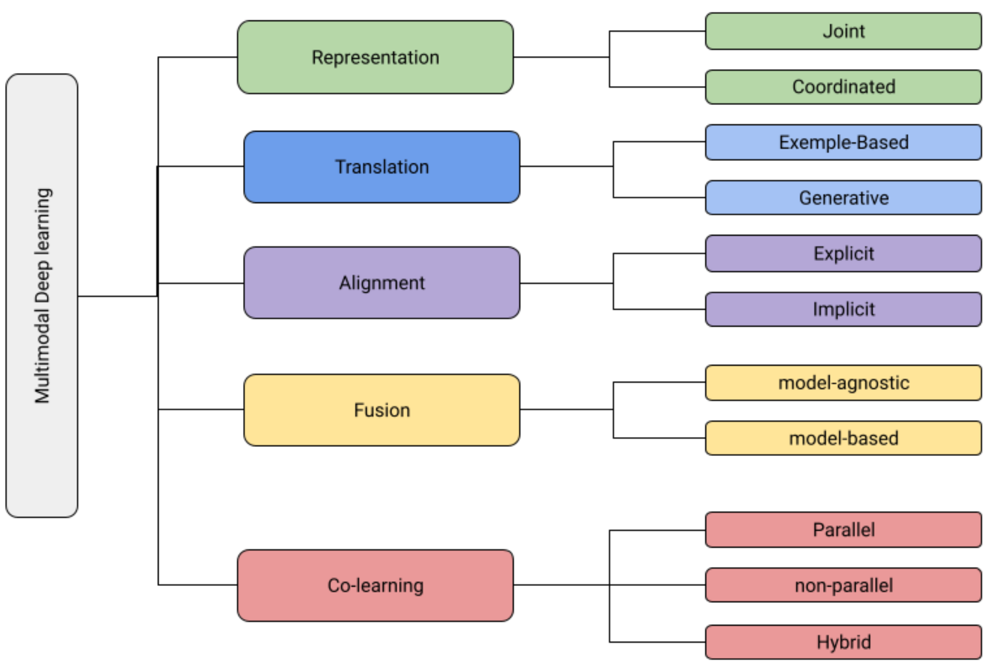


Figure 3. 1 Multimodal deep Learning problems(Barua et al., 2023)

2.2. Representation

Representation in multimodal deep learning involves encoding information from various modalities into a unified format that models can process. The goal is to capture the complementary information from each modality to improve learning and inference. Effective representation is crucial for enabling models to understand and synthesize information from different sources(Baltrušaitis et al., 2017; Barua et al., 2023).

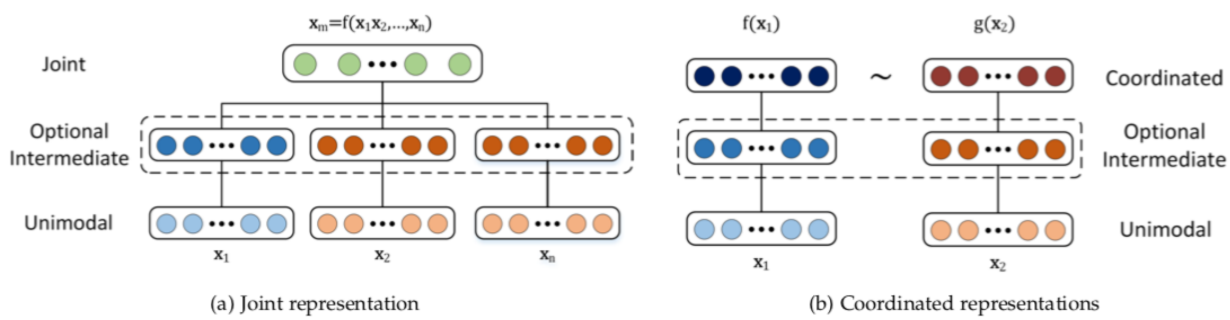


Figure 3. 2 Structures of joint and coordinated representations (Structure Of Joint And Coordinated Representations, s. d.)

2.2.1. Joint Representations

Joint representations involve combining information from different modalities into a single unified representation. This is typically done through concatenation or transformation techniques that merge features from different modalities, allowing the model to leverage information across these sources simultaneously. By doing so, joint representations can capture the complex interactions and correlations between modalities, providing a comprehensive understanding of the data.

2.2.2. Coordinated Representations

Coordinated representations keep the representations of different modalities separate but aligned in such a way that they can be effectively used together. Techniques like canonical correlation analysis (CCA) are often used to ensure that the separate modality-specific representations are correlated. This approach maintains the integrity of each modality while still enabling effective multimodal integration, facilitating tasks that require a nuanced understanding of each modality's contributions.

2.3. Translation

Translation models aim to transform data from one modality into another. For instance, transforming an image into a textual description or vice versa (as illustrated in Figure 3.3). This requires models to understand the semantics of both modalities and find a common ground for accurate translation (Baltrušaitis et al., 2017; Elliott & Kádár, 2017)

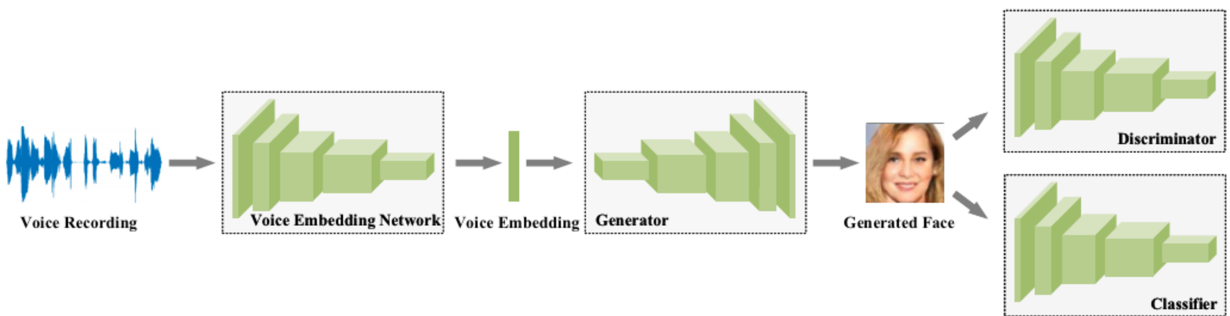


Figure 3. 3 A model that generates faces from voices (Mercari Engineering, 2021)

2.3.1. Example-based Methods

Example-based methods use a dataset of paired examples (e.g., image-caption pairs) to learn the relationships between modalities. The model learns by directly comparing new inputs to the stored examples to find the closest match. This approach relies on a rich database of examples to ensure accurate translation, leveraging the direct correlations found within the paired data.

2.3.2. Generation Approaches

Generation approaches involve creating new data in one modality based on the input from another modality. Examples include generating a textual story from an image or synthesizing an image based on a text description. These approaches require sophisticated models that can not only understand the input modality but also generate coherent and contextually relevant output in the target modality. Encoder-decoder models are often employed in this context to handle the complex transformation processes involved in generative tasks.

2.4. Alignment

Alignment involves ensuring that corresponding elements from different modalities are properly matched. This can be done in various ways (Baltrušaitis et al., 2017; Rasenberg et al., 2020):

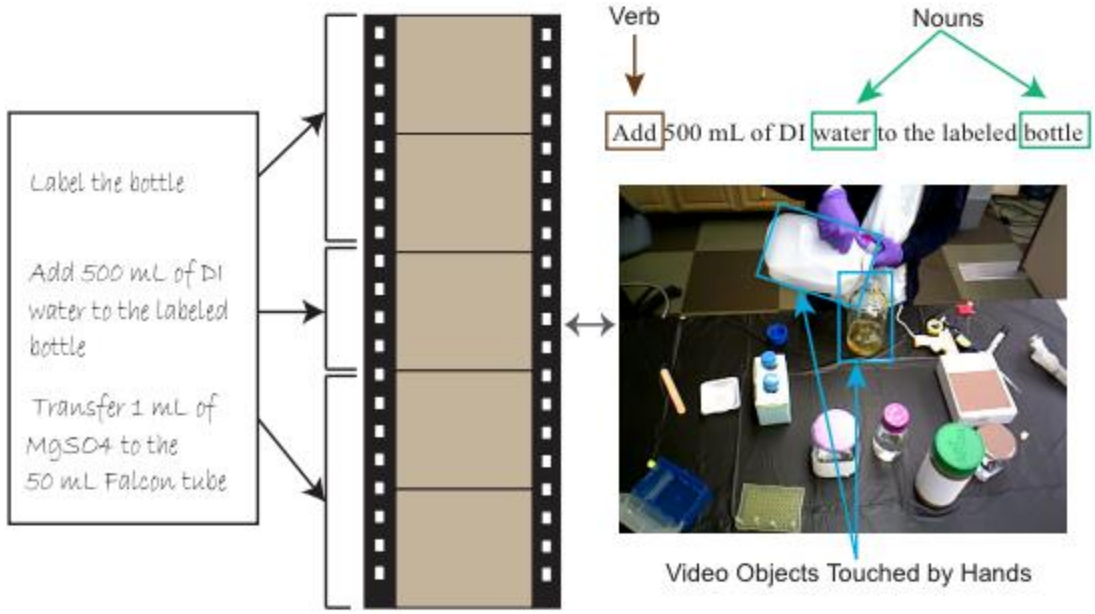


Figure 3. 4 Alignment of video segments with text sentences (Vinija's Notes • Multimodal Machine Learning • Multimodal Challenges, s. d.)

2.4.1. Explicit Alignment

Explicit alignment uses labeled data to directly learn the correspondences between modalities. Supervised learning techniques are often employed here, relying on annotated datasets to train models to map elements from one modality to their counterparts in another. This method is powerful but depends heavily on the availability and quality of labeled data.

2.4.2. Implicit Alignment

Implicit alignment relies on models to learn the correspondences between modalities indirectly through the training process, without explicit supervision. Unsupervised methods such as clustering and dimensionality reduction are commonly used. These techniques identify statistical relationships within the data, allowing models to infer alignments without needing explicit labels.

2.5. Fusion

Fusion in multimodal deep learning refers to the process of combining information from multiple modalities to improve learning and prediction. This challenge is crucial because each modality can provide unique and complementary information that, when integrated, can lead to more robust and accurate models. There are several types of fusion techniques, broadly categorized into model-agnostic and model-based methods(Atrey et al., 2010; Baltrušaitis et al., 2017; Boulahia et al., 2021).

2.5.1. Model Agnostic

Model agnostic fusion techniques are versatile and can be applied to any type of model. They do not rely on the specific architecture of the underlying models, making them flexible and widely applicable.

A. Early Fusion

Early fusion, also known as feature-level fusion, involves combining raw data from different modalities before feeding it into the model. This approach allows the model to learn low-level correlations between modalities(Baltrušaitis et al., 2017), which can be beneficial for tasks requiring integrated feature representations. However, early fusion can lead to high-dimensional data, which may pose challenges in terms of computational complexity and the risk of overfitting.

B. Intermediate Fusion

Intermediate fusion is a technique that combines features from different modalities by integrating features from each modality at a middle layer (Boulahia et al., 2021); it creates a unified representation that leverages the strengths of all input types. This method can enhance the expressive power of the data representation, often leading to improved performance in recognition tasks compared to using each modality in isolation

C. Late Fusion

Late fusion, also known as decision-level fusion, combines the outputs of modality-specific models at a later stage. This approach processes each modality independently before integrating the final predictions. Late fusion allows each modality to be modeled separately, which can be advantageous when dealing with noisy or heterogeneous data. It is often more robust to individual modality errors, as each modality's influence can be weighted appropriately in the final decision(Baltrušaitis et al., 2017).

2.5.2. Model-based Methods

Model-based fusion methods build explicit models to understand and combine multimodal data. These methods often incorporate domain knowledge and structured representations to enhance the fusion process(Baltrušaitis et al., 2017).

A. Kernel-based Methods

Kernel-based methods use kernel functions to map data into higher-dimensional spaces, where the relationships between different modalities can be more easily captured and analyzed. By transforming the data, these methods can uncover complex, non-linear interactions that might not be apparent in the original feature space.

B. Graphical Models

Graphical models use graph structures to represent and reason about the dependencies between different variables in multimodal data. Examples include Bayesian networks and Markov random fields, which can model the probabilistic relationships between modalities. These models can effectively capture the interdependencies and conditional independencies within multimodal data, facilitating robust fusion.

C. Neural Networks

Neural networks are widely used in multimodal fusion due to their ability to learn complex patterns and representations. Different architectures, such as convolutional neural networks (CNNs) for images and recurrent neural networks (RNNs) for text, are often combined.

These networks can be designed to fuse data at various stages, from early to late fusion, depending on the task requirements.

D. Singular Value Decomposition (SVD)

Singular Value Decomposition (SVD) is a matrix factorization technique used to reduce the dimensionality of multimodal data while preserving important features. It can be used for tasks like latent semantic analysis, where the goal is to uncover the underlying structure in the data. SVD helps in managing the high-dimensionality of fused data and improves computational efficiency.

2.6. Co-learning

Co-learning involves training models using multiple data sources to enhance performance by leveraging the strengths of each modality. This approach is beneficial even when some modalities are not always available, are noisy, or are in limited amounts. Below, we see the three sub-categories of co-learning, as depicted in Figure 3.5 (Rahate et al., 2022).

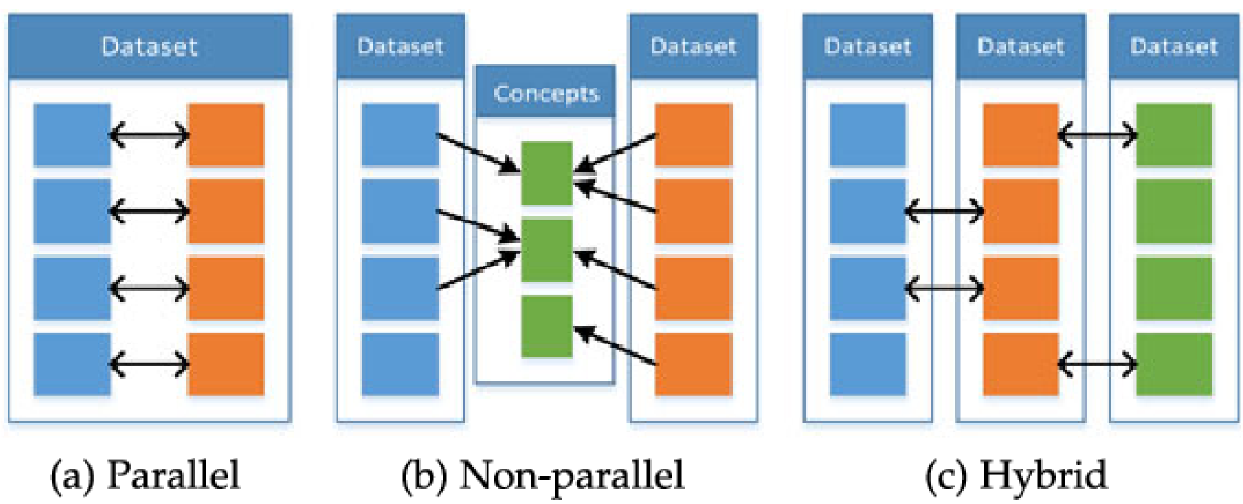


Figure 3. 5 Categories of co-learning (Mercari Engineering, 2021)

2.6.1. Parallel

Parallel data consists of modalities that are directly aligned, such as text and audio in a speech dataset. Techniques include co-training and translation-based co-learning, where paired data like image-caption pairs are used to enhance the model's learning capabilities(Rahate et al., 2022)

2.6.2. Non-parallel

Non-parallel data involves modalities that are not directly aligned but share a common context, like instructional videos with audio and text. Techniques include using attention models and multimodal embeddings to find relationships between different modalities, as seen in models like DeViSE, which align text and images through shared semantic spaces(Baltrušaitis et al., 2017; Rahate et al., 2022)

2.6.3. Hybrid Data

Hybrid data uses a shared modality to bridge other modalities, such as using images as pivots for multilingual text in image captioning. Techniques include using shared relationships and multimodal transformers to align various data types, enabling tasks like multilingual video search(Rahate et al., 2022).

2.7. Conclusion

In summary, multimodal deep learning leverages diverse sensory modalities to enhance learning and inference across various tasks. By addressing challenges in representation, translation, alignment, fusion, and co-learning, researchers can develop models that effectively integrate and process multimodal data, leading to improved performance and broader application capabilities in real-world scenarios

3. Design and implementation

3.1. Introduction

Visual Question Answering (VQA) faces challenges due to biases in datasets, prompting the creation of the CLEVR dataset to ensure rigorous testing of reasoning abilities with 3D-rendered scenes and complex questions. Additionally, the VQA-RAD dataset, focused on radiology, pairs medical images with clinically relevant questions to aid decision support. This chapter explores models and fusion techniques to enhance VQA performance on these datasets, aiming to identify the most effective combinations for robust multimodal systems.

3.2. Datasets

3.2.1. CLEVR dataset

Many Visual Question Answering (VQA) datasets have biases that let models find shortcuts without really understanding the visual input. The CLEVR dataset was created to fix this and focus on testing true reasoning abilities. CLEVR has images of 3D-rendered objects, each with a variety of detailed questions. These questions are grouped into five types: Exist, Count, Compare Integer, Query Attribute, and Compare Attribute. The dataset includes(Johnson et al., 2016; Marois et al., 2018):

- 70,000 training images with 700,000 questions,
- 15,000 validation images with 150,000 questions,
- 15,000 test images with 150,000 questions.

Each object in the images is described by its position and four attributes:

- Size: large or small,
- Shape: square, cylinder, or sphere,
- Material: rubber or metal,
- Color: gray, blue, brown, yellow, red, green, purple, or cyan.

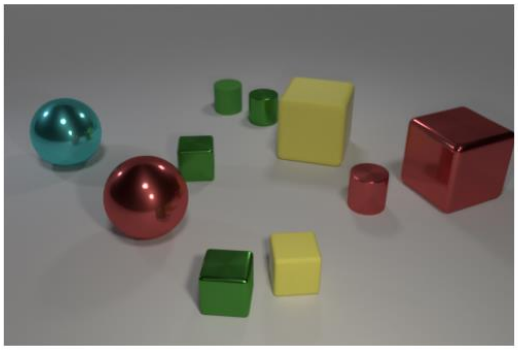
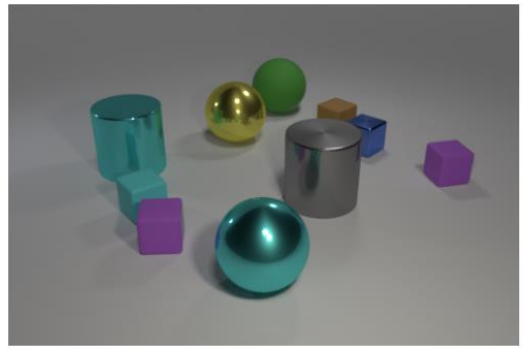
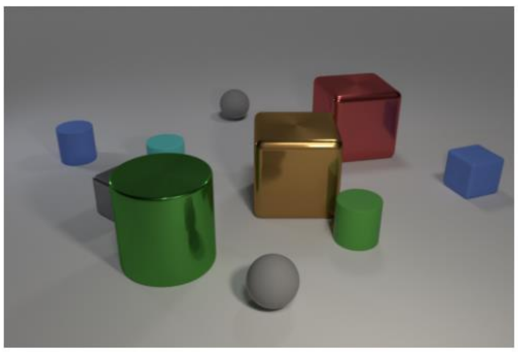
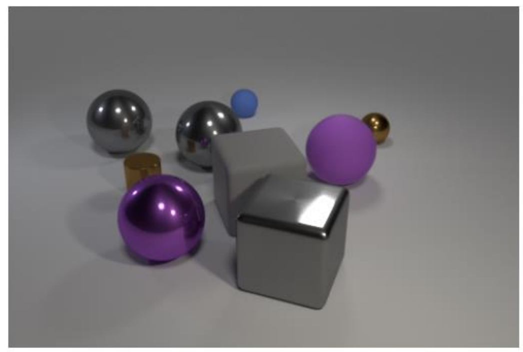
 Q: What number of other things are the same material as the red cube?	 Q: What material is the tiny cube that is to the right of the blue thing?
 Q: What size is the thing that is the same color as the large metallic cylinder?	 Q: Is the color of the big matte ball the same as the tiny rubber ball?

Figure 4. 1 Example of clever dataset

3.2.2. VQA-RAD dataset

The VQA-RAD dataset is a manually constructed dataset in the radiology domain. This dataset aims to aid in clinical decision support by providing a structured collection of radiology images paired with clinically relevant questions and their corresponding answers. Here are the key points about the VQA-RAD dataset:(Lau et al., 2018)

Images: 315 radiology images sourced from MedPix®, an open-access radiology archive.

- Questions and Answers: A total of 3,515 visual questions, with 1,515 being free-form and 733 of those having corresponding paraphrased versions. These questions are manually created by clinicians to reflect natural clinical inquiries.
- Question Types: Categorized into types such as abnormality detection, object/condition presence, positional reasoning, color, size, attribute description, counting, and others.
- Answer Types: Includes both close-ended (e.g., yes/no) and open-ended questions.

The quality and validation of the VQA-RAD dataset are ensured by:

Manual Validation: All question-answer pairs are manually validated and categorized by clinicians.

Clinical Relevance and Accuracy: Naturally occurring questions are used, rather than automatically generated ones, which ensures a high level of clinical relevance and accuracy.

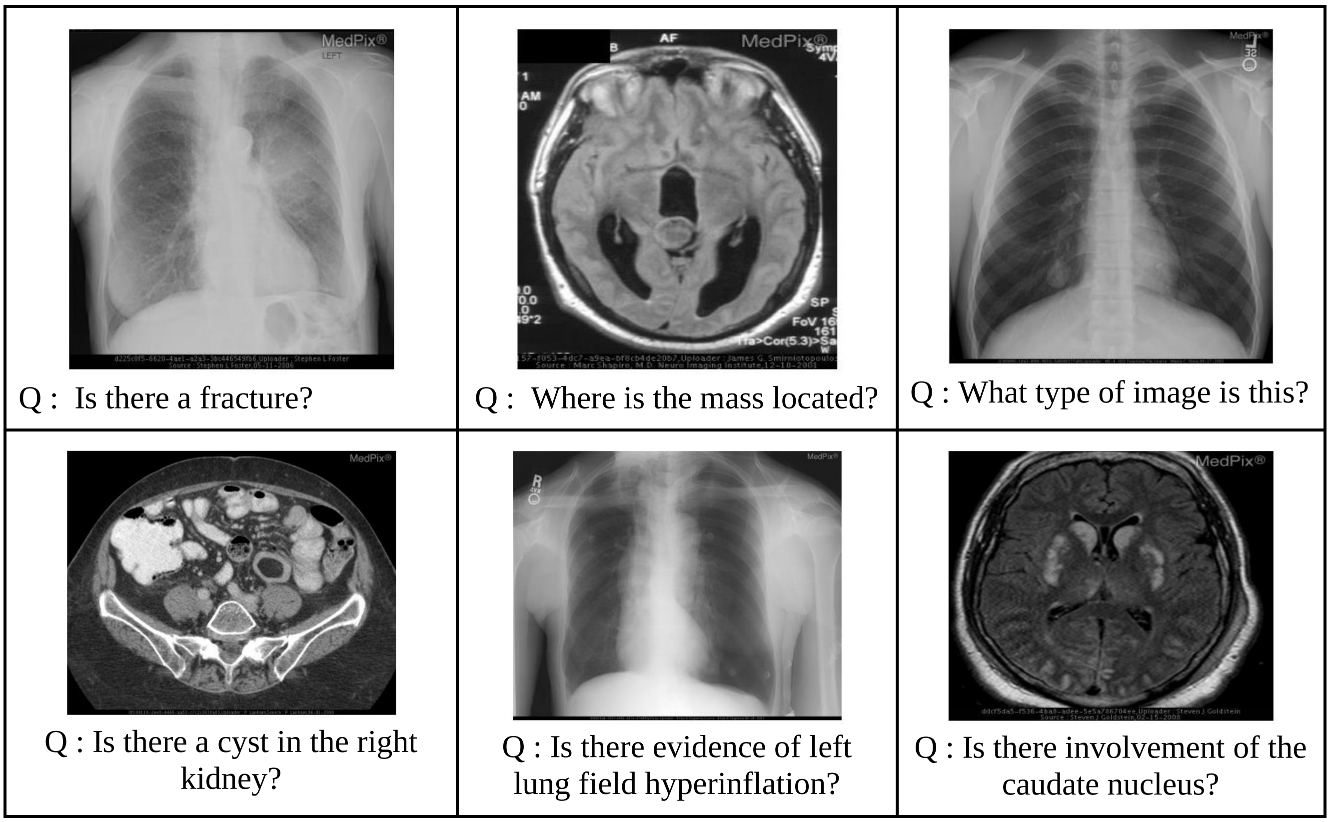


Figure 4. 2 example of VQA-RAD

3.3. Pre-processing

3.3.1. Image pre-processing

A. Image Loading and Conversion

The image is loaded using Image.open() and converted to RGB mode with convert('RGB'). This standardizes all images to three color channels.

B. Resizing

The image is resized to 224x224 pixels using transforms.Resize(). This resizing is essential because many convolutional neural networks (CNNs), require inputs of a specific size. Standardizing image dimensions ensures consistency and compatibility with the model architecture, which is designed to handle inputs of 224x224 pixels.

C. Normalization

The image is converted to a tensor with transforms.ToTensor(), scaling pixel values to [0, 1]. It is then normalized using transforms.Normalize() with mean [0.485, 0.456, 0.406] and standard deviation [0.229, 0.224, 0.225], which are based on the ImageNet dataset. This ensures consistency with the model's training data.

D. Batch Dimension Addition

A batch dimension is added using torch.unsqueeze(), changing the tensor shape (Channels, Height, Width) to [1, C, H, W]. This prepares the image for batch processing by the model.

3.3.2. Question pre-processing

A. **Tokenization**

For BERT, `tokenizer.tokenize()` from BertTokenizer is utilized to break the text into words or subwords. For the LSTM model, NLTK's `word_tokenize()` is used to split the text into individual words.

B. **Lemmatisation**

Lemmatisation reduces words to their base form using NLTK's WordNetLemmatizer. This step normalizes the text and reduces morphological variations, enhancing model understanding.

C. **Special Tokens Addition** (for BERT)

For BERT, special tokens are added to the token list to denote the beginning and end of the sequence. The `[CLS]` token is placed at the beginning of the token sequence and the `[SEP]` token is added at the end.

D. **Conversion to Input IDs**

Tokens are converted into input IDs using `tokenizer.convert_tokens_to_ids()` for BERT. For the LSTM model, a vocabulary is created by assigning each unique token an ID, and tokens are then converted to their corresponding IDs based on this vocabulary.

E. **Padding and Truncation**

To ensure all input sequences are of a fixed length, sequences are padded or truncated to a specified maximum sequence length (e.g., 50 tokens). Padding involves adding zeroes (0) to the end of shorter sequences, while truncation involves cutting off longer sequences to fit the maximum length.

F. **Conversion to Tensor**

Input IDs are converted into a PyTorch tensor using `torch.tensor()`, and a batch dimension is added using `unsqueeze(0)`. This transformation converts the input IDs from a one-dimensional array to a two-dimensional tensor with shape [1, max_seq_length].

G. **Attention Mask Creation** (for BERT):

An attention mask is created to distinguish between real tokens and padding tokens. The attention mask is a tensor of the same shape as the input IDs tensor, with ones (1) for real tokens and zeroes (0) for padding tokens.

H. **Embedding Layer** (for LSTM with Word2Vec):

For LSTM and models, pre-trained Word2Vec embeddings are used to convert input IDs into dense vector representations. The embedding matrix is created by loading pre-trained Word2Vec embeddings, where each row corresponds to a word in the vocabulary, and the values represent the embedding vector for that word.

I. **Forward Pass through Models**:

BERT: The preprocessed input IDs and attention mask are fed into the BERT model. The last hidden state corresponding to the `[CLS]` token is extracted and used as the representation of the input question. A linear transformation via a fully connected layer is applied to obtain the final question representation.

LSTM: The embedded inputs are fed through the LSTM model. The output from the last hidden state or the final sequence representation is used as the representation of the input question.

3.3.3. Answer pre-processing

A. **Defining Vocabulary**

Extract unique labels (answers) from the DataFrame column and store them in a set to ensure uniqueness.

B. **Creating a Word-to-Index Dictionary**

Construct a dictionary to map each unique answer to a unique index. This mapping facilitates the conversion of words to numerical representations.

C. **One-Hot Encoding**

Retrieve the index of the word from the word-to-index dictionary and create a one-hot vector with the corresponding index set to 1 and all other elements set to 0. Convert the resulting one-hot vector to a PyTorch tensor.

3.4. Design and construction of the model

3.4.1. Methodology

Image Feature Extraction :

Load the model with pre-trained weights.

Retain only the feature extraction layers.

Flatten the output to a 1D vector.

Pass the features through a dense layer to obtain the final feature representation.

Question Feature Extraction:

Load the BERT model.

Use only the feature extraction part of BERT, focusing on the [CLS] token representation.

Flatten the [CLS] token output.

Pass the features through a dense layer to match the size of image features.

VQA Model :

Combine the image and question features using element-wise multiplication.

Pass the combined features through a fully connected layer.

Include a dropout layer for regularization.

Apply a softmax activation function for classification.

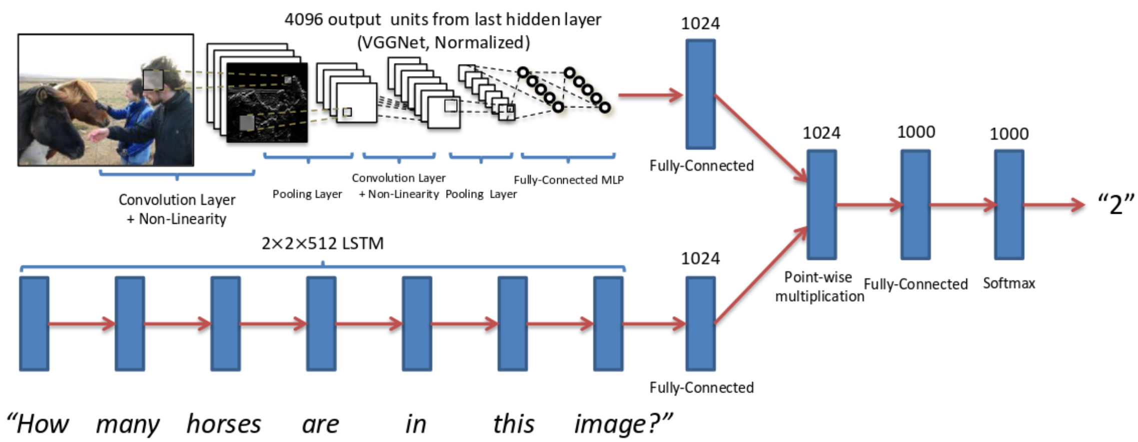


Figure 4. 3 visual question answering system (Franky, 2021)

3.4.2. Training phase

Define the Loss Function and Optimization Algorithm

The loss function used is CrossEntropyLoss, suitable for multi-class classification tasks. The Adam optimizer is employed to adjust the learning rate based on the gradients' first and second moments, facilitating efficient training.

Incorporate a Learning Rate Scheduler

The StepLR scheduler is incorporated to adjust the learning rate throughout training. The learning rate is reduced by a factor of 0.1 every 7 epochs, allowing finer control as training progresses.

Split the Dataset into Training and Validation Sets

The dataset is divided into training and validation sets using an 80-20 ratio, meaning 80% of the data is used for training, and 20% is reserved for validation. This ensures the model is trained on a substantial portion of the data while being validated on a separate set to monitor performance.

Extract Image and Question Features

Image features are extracted using a pre-trained model, and question features are extracted using a BERT and lstm model. The pre-trained models are utilized to leverage their learned representations, providing robust feature extraction for both image and textual data.

Weight Initialization

The Xavier initialization method is used to enhance performance by setting the initial weights, helping maintain variance and preventing issues like vanishing or exploding gradients.

3.5. Evaluate Model Performance

3.5.1. Metrics

Confusion Matrix: A confusion matrix is a table used to evaluate the performance of a classification algorithm. It displays the number of true positives (TP), true negatives (TN), false positives (FP), and false negatives (FN).

Accuracy: Accuracy is a measure of the overall correctness of a classification model. It is the ratio of correctly predicted instances (both true positives and true negatives) to the total number of instances. $Accuracy = \frac{TP+TN}{TP+TN+FP+FN}$

Precision: Precision (also called Positive Predictive Value) measures the accuracy of the positive predictions. It is the ratio of true positives to the total number of positive predictions (true positives plus false positives). $Precision = \frac{TP}{TP+FP}$

Recall: Recall (also called Sensitivity or True Positive Rate) measures the ability of the model to identify all relevant instances. It is the ratio of true positives to the total number of actual positives (true positives plus false negatives). $Recall = \frac{TP}{TP+FN}$

F1 score: The F1 Score is the harmonic mean of precision and recall, providing a single metric that balances both concerns. It is useful when you need a balance between precision and recall, especially in cases where you have an uneven class distribution. $F1\ Score = 2 \times \frac{Precision \times Recall}{Precision+Recall}$

3.5.2. Experimental setup

the code was executed by the laptop with specific configuration:

- Processor: 11th Intel Core (TM) i5-11400H @2.70GHz

- RAM: 16 GB.

- Graphics Card: NVIDIA GeForce RTX 3050 with 4 GB of dedicated memory

- Storage: 512 To SSD drive

The experiments were performed using the Python programming language, leveraging the PyTorch library for deep learning model development and evaluation

3.5.3. Results and discussions

In this experimental study, we explore a range of models and methodologies to enhance the performance of visual question answering (VQA) tasks using the CLEVR dataset. Our approach involves leveraging various state-of-the-art image feature extraction models, including VGG19, ResNet50, EfficientNet-B5, and Vision Transformer (ViT). For the extraction of question features, we utilize BERT and BiLSTM models. Additionally, we investigate the effectiveness of two fusion operations multiplication and concatenation integrating these features to build robust VQA systems.

By comparing these diverse configurations, we aim to identify the optimal combination of image and question feature extractors and fusion methods that achieve the highest accuracy and generalization capabilities. The results of our study provide insights into the performance and efficacy of these models and fusion techniques, contributing valuable knowledge to the field of multimodal deep learning.

Here, we present the findings of our experiments conducted on the CLEVR dataset, detailing the performance metrics and observations for each configuration.

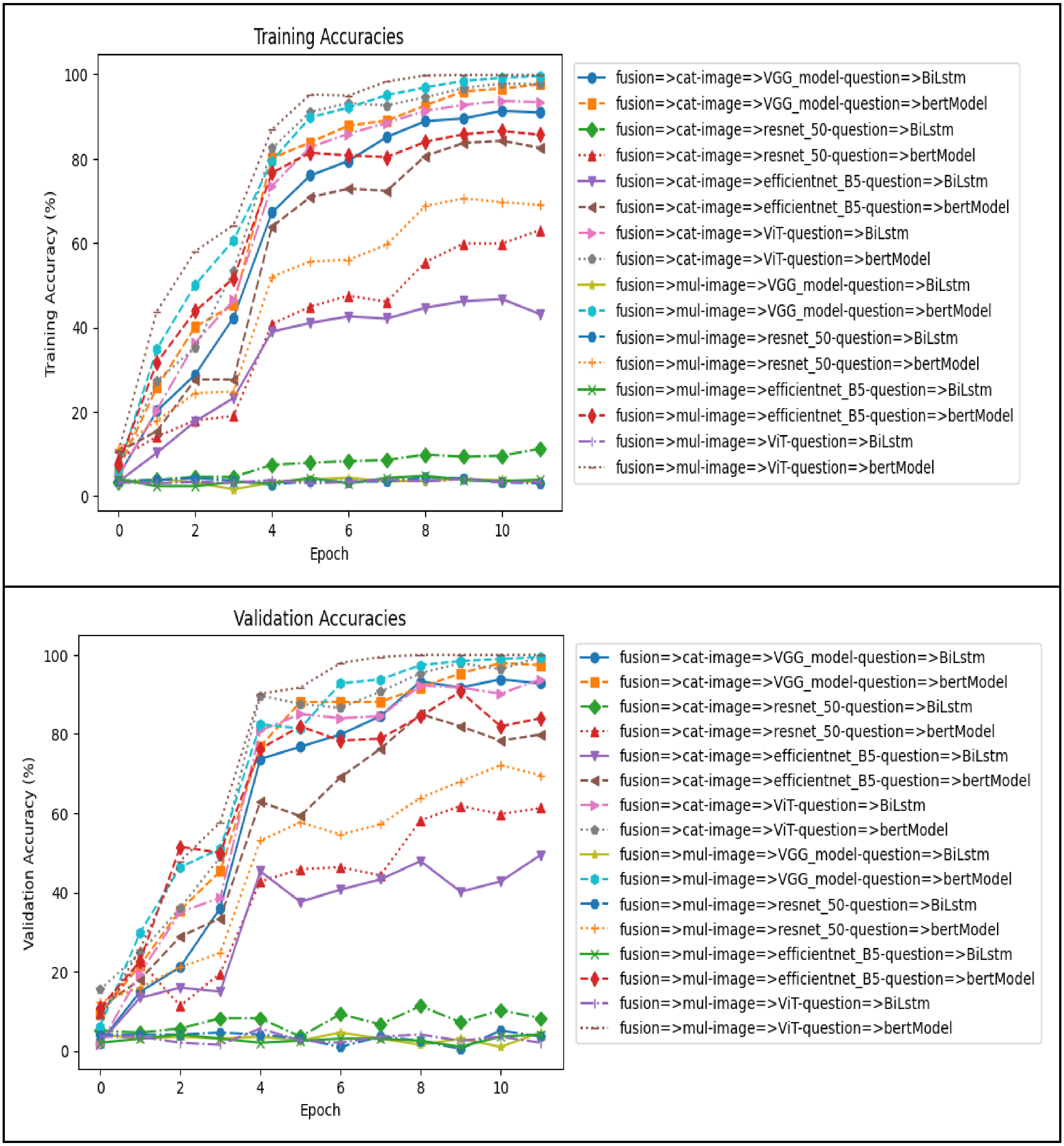


Figure 4. 4 Comparison of accuracies curves for train and validation

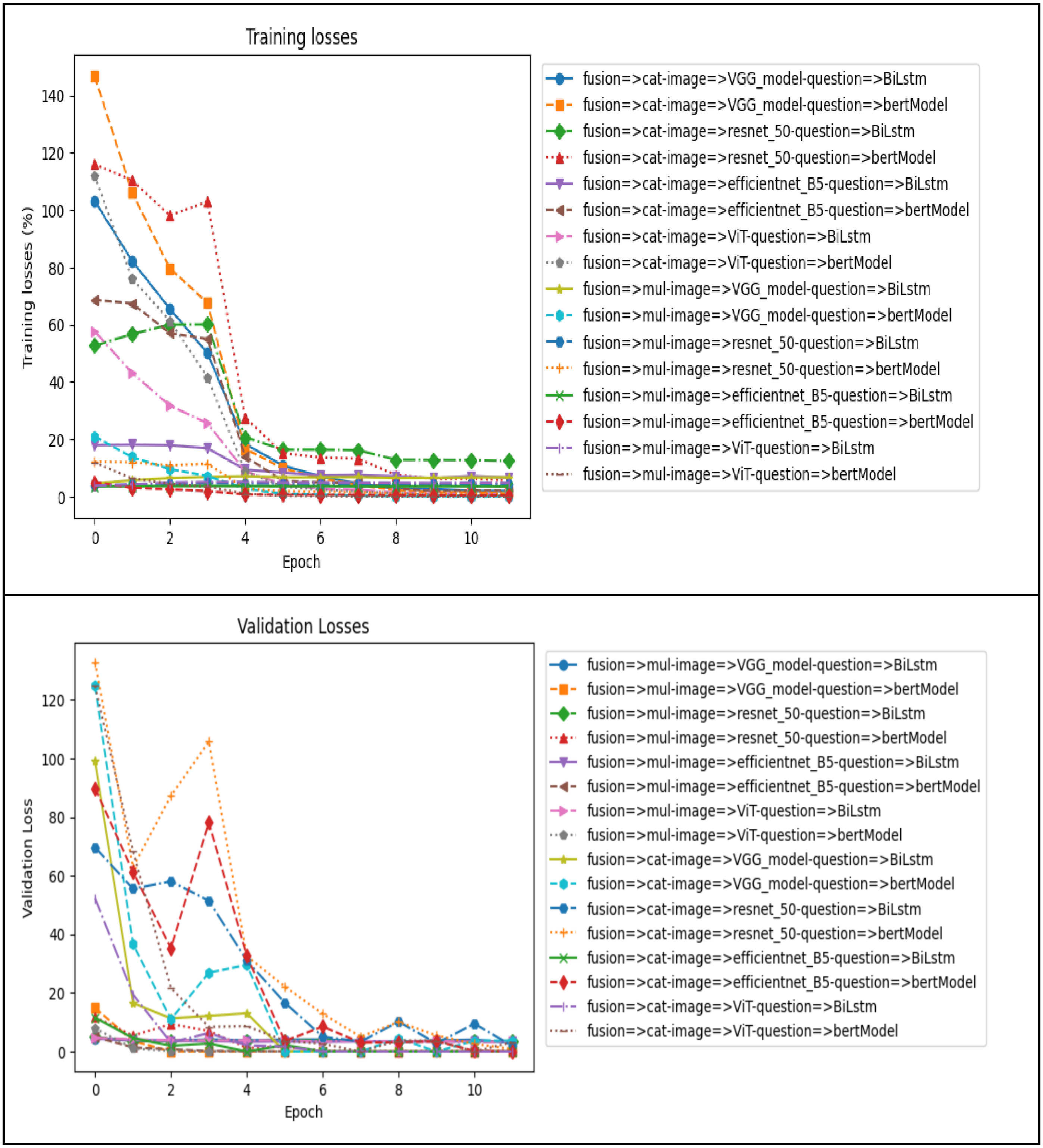


Figure 4. 5 Comparison of losses curves for train and validation

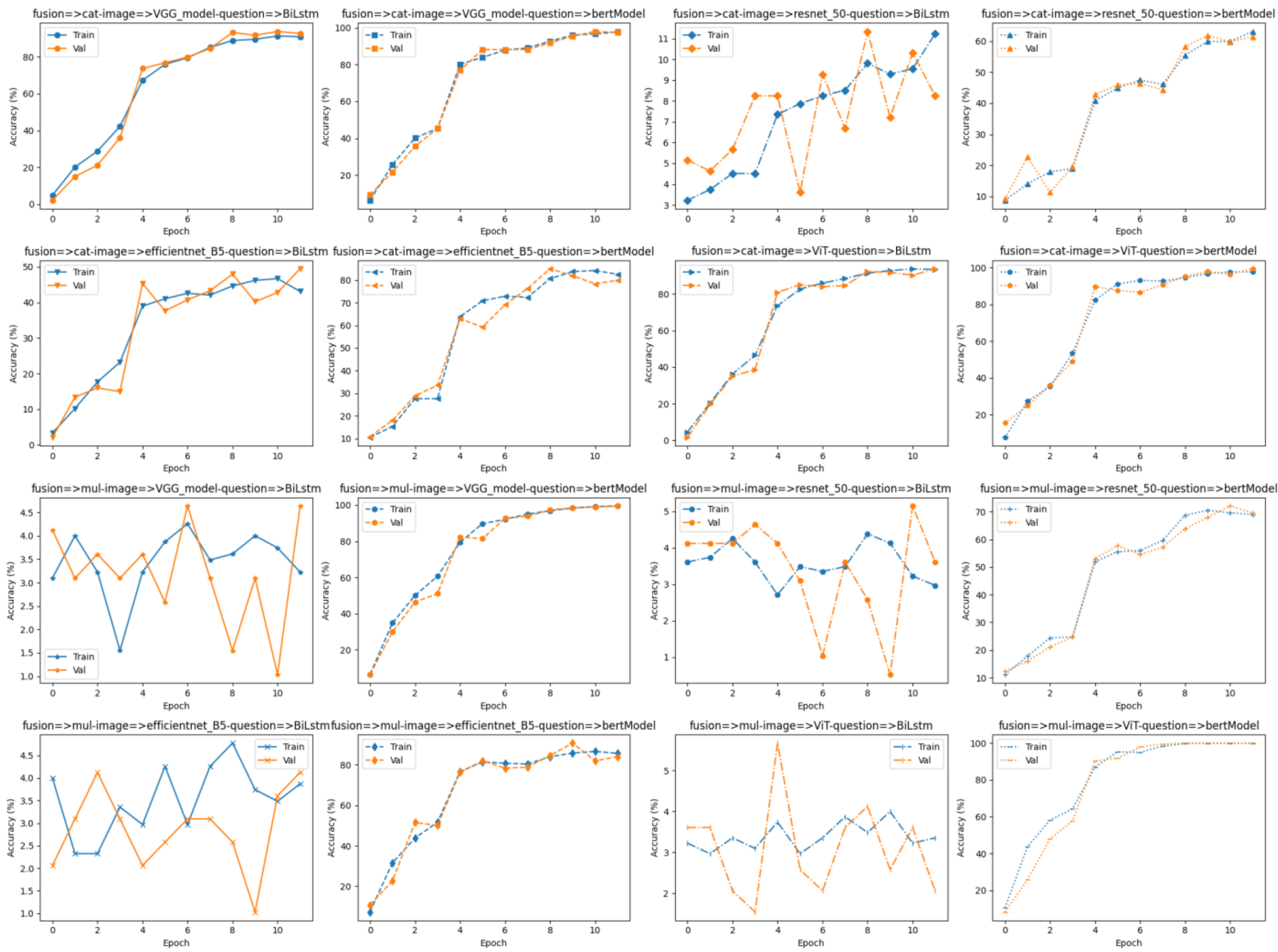


Figure 4. 6 Performance Metrics of Fusion Models with Different Architectures

Discussions

Top Performers :

1. Fusion=>mul, image=>ViT, question=>bertModel:
   - Achieves high validation accuracy early on, reaching 98.7% and maintaining it.
   - Indicates that the Vision Transformer (ViT) combined with BERT and the multiplication fusion method is highly effective for the CLEVR VQA task.
2. Fusion=>mul, image=>VGG_model, question=>bertModel:

  - Also reaches high validation accuracy, performing slightly below the ViT and BERT combination but still among the best performers. with the accuracy of 98.2%
  - Suggests that VGG19 with BERT using the multiplication fusion method is a strong configuration.

Observations :

1. Fusion=>mul, image=>ViT=>question, bertModel:
   - Training and validation accuracies are closely aligned, demonstrating excellent generalization.
   - The high and stable performance across epochs indicates that this model is highly effective for the task.
2. Fusion=>mul, image=>VGG_model, question=>bertModel:
   - Shows consistent performance with training and validation accuracies closely matching.
   - This combination also generalizes well, maintaining high accuracy across epochs.

Model Performance Analysis

1. Image Model Influence :
   - ViT (Vision Transformer): Performs exceptionally well, likely due to its advanced architecture that captures detailed image features effectively.
   - VGG19: Also performs strongly, benefiting from its well-established feature extraction capabilities.
2. Question Model Influence :
   - BERT: Outperforms BiLSTM in all cases, demonstrating its superior ability to understand and encode questions contextually.
3. Fusion Method :
   - Multiplication (mul): Surprisingly effective when combined with ViT or VGG19 and BERT, leading to top performance.
   - Concatenation (cat): Generally, performs well but not as consistently as the multiplication method in the top-performing configurations.

Result

- The combination of multiplication fusion with ViT for images and BERT for questions provides the best performance on the CLEVR dataset.
- The multiplication fusion with VGG19 for images and BERT for questions also performs very well, slightly below the top combination.

- This suggests that the multiplication fusion method, when combined with powerful image models like ViT and robust question models like BERT, is highly effective for VQA tasks.
- The general stability and high accuracy of these models across epochs indicate their strong generalization capabilities and robustness.

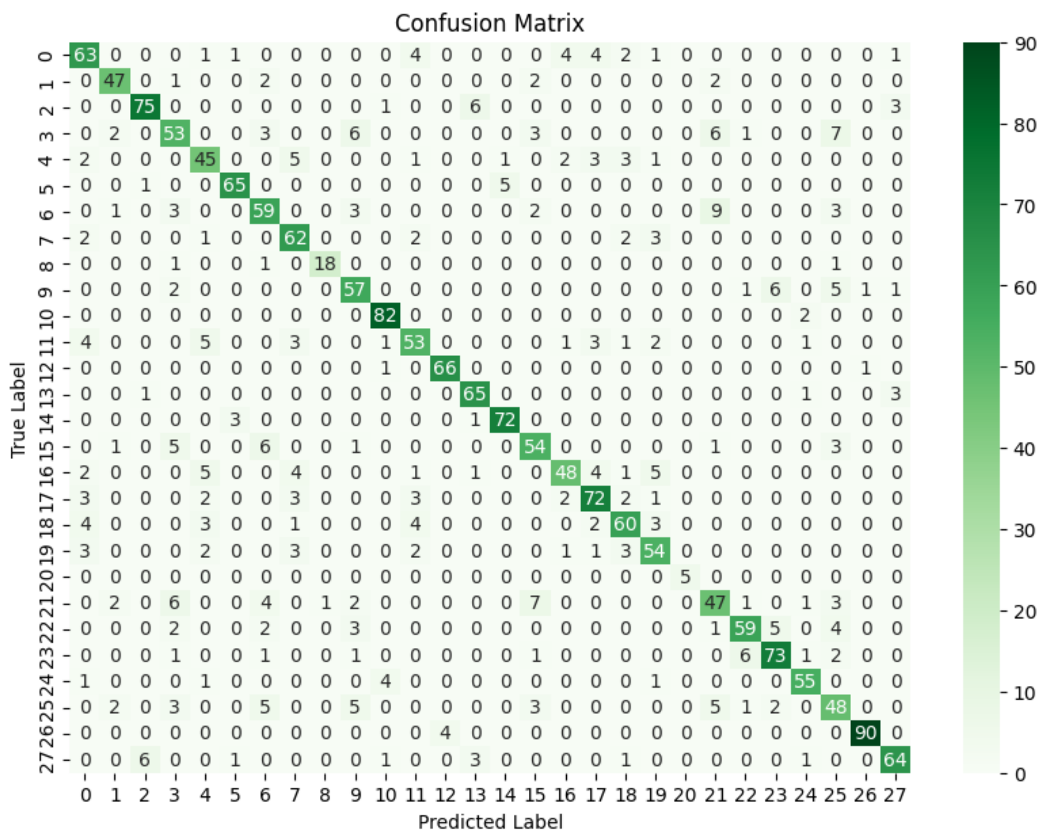
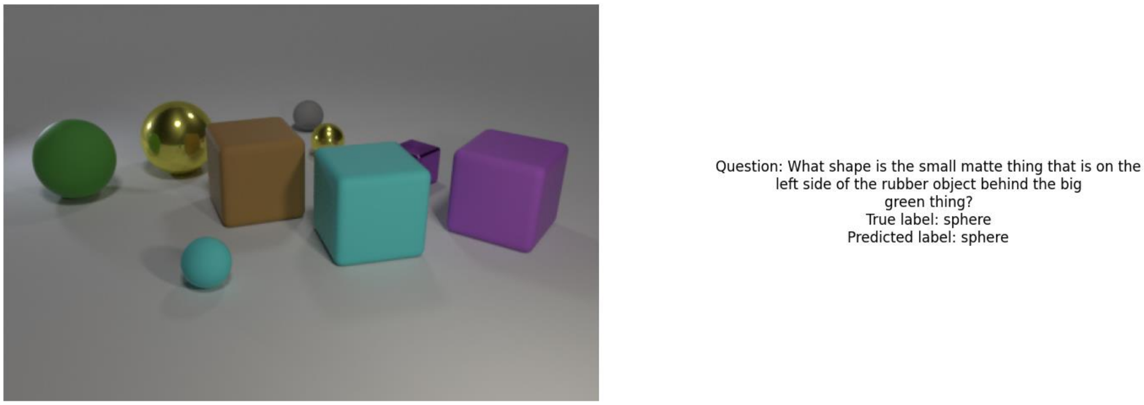
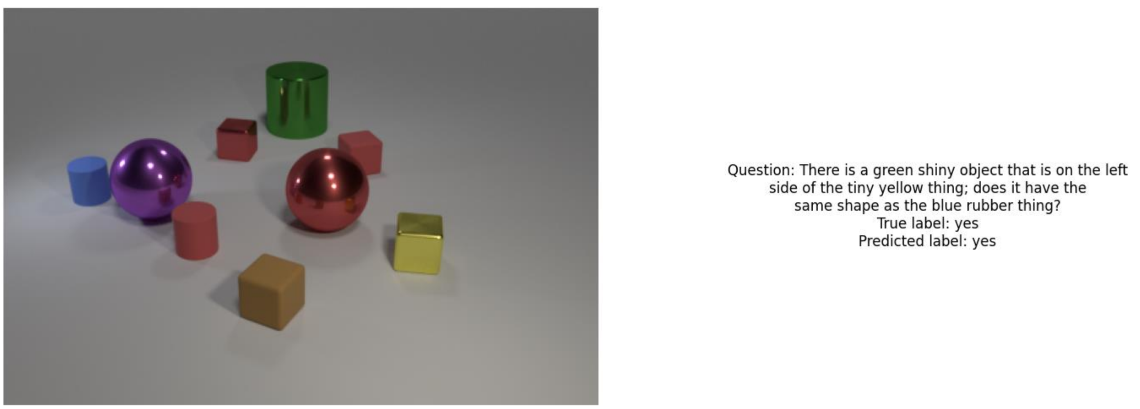


Figure 4. 7 Confusion matric on clevr dataset

The confusion matrix generated by the top model shows strong diagonal dominance, indicating high accuracy for most classes. Off-diagonal values highlight misclassifications, revealing specific challenges like confusing similar categories such as numbers (1 and 0) or materials (rubber vs metal)

Result exemples





VQA-RAD dataset

In this section, we evaluate the performance of the best model identified from our initial experiments on a real-world dataset. Specifically, we utilize the VQA-RAD dataset, which comprises medical images and associated questions, to test the effectiveness of our selected model.

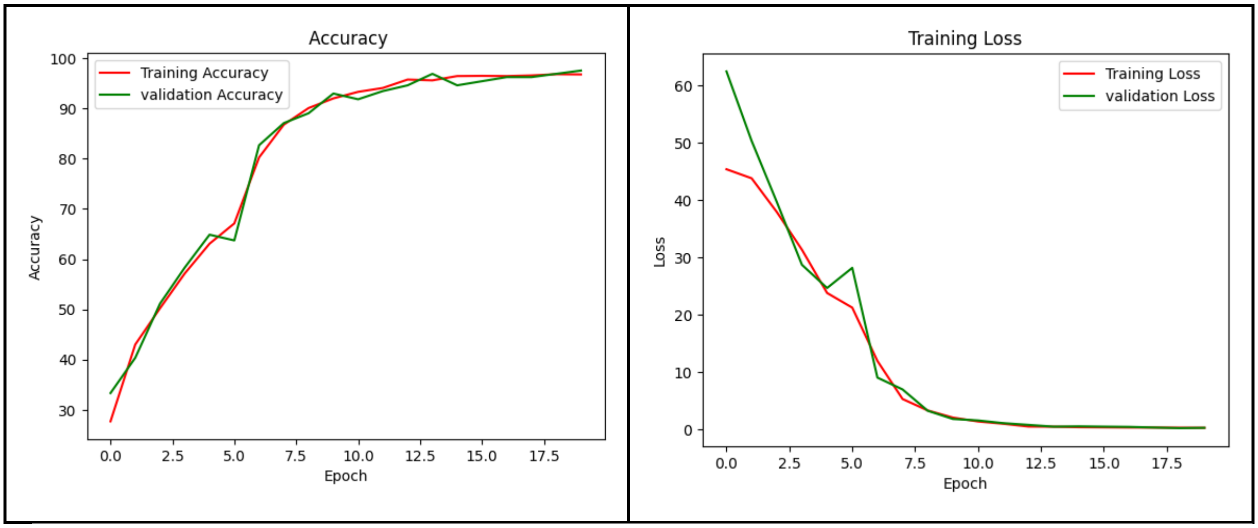


Figure 4. 8 Training and validation accuracy/loss curves for Vit-bert-mul on VQA-RAD dataset

The model achieved an accuracy of 96%. Precision, recall, and F1 score are all 0.96, reflecting balanced performance with low false positive and false negative rates, and a good balance between precision and recall. The accuracy curve shows that the training and validation accuracy Curves grow gradually and follow each other closely. suggesting effective learning without significant overfitting or underfitting. The loss curve demonstrates that both training and validation loss decrease steadily and converge, indicating that the model is learning effectively and generalizing well to the validation data. Here is example result

Question: Are there any abnormalities in the kidneys?
True label: No
Predicted label: No

Question: What type of imaging does this not represent?
True label: ultrasound
Predicted label: ultrasound

3.6. Conclusion

In this experimental study, we conducted a comprehensive evaluation of various image and question feature extraction models and fusion methods for Visual Question Answering tasks using the CLEVR dataset. Our results indicate that the combination of Vision Transformer (ViT) for image features and BERT for question features, integrated using the multiplication fusion method, achieves the highest accuracy, reaching 98.7%. This configuration demonstrates excellent generalization and robustness, maintaining high and stable performance across epochs. Similarly, the combination of VGG19 and BERT with multiplication fusion also performs strongly, achieving an accuracy of 98.2%.

Additionally, we validated the effectiveness of our best-performing model on the VQA-RAD dataset, which includes medical images and clinically relevant questions. The model achieved an accuracy of 96%, with balanced precision, recall, and F1 scores, reflecting its strong generalization capabilities and suitability for real-world applications.

Overall, our findings suggest that the multiplication fusion method, when combined with advanced image models like ViT and robust question models like BERT, provides a highly effective approach for VQA tasks.

4. General Conclusion

In this work, we explored key aspects of machine learning and deep learning, focusing on foundational models such as Convolutional Neural Networks (CNNs) and Transformers. These models are crucial for advancements in computer vision and natural language processing, underpinning complex applications like Visual Question Answering (VQA). VQA requires the integration of visual and textual data to produce relevant responses.

Our experimental studies conducted an in-depth evaluation of various image and question feature extraction models and fusion methods for VQA tasks using the CLEVR dataset. We found that combining the Vision Transformer (ViT) for image features with BERT for question features, using the multiplication fusion method, achieved the highest accuracy of 98.7%. This model showed excellent generalization and robustness. Similarly, the combination of VGG19 and BERT with multiplication fusion also performed well, achieving an accuracy of 98.2%.

To extend our findings, we tested our top-performing model on the VQA-RAD dataset, which includes medical images and clinical questions. The model attained an accuracy of 96%, with balanced precision, recall, and F1 scores, highlighting its potential for real-world applications.

Our research suggests that advanced image models like ViT and robust question models like BERT, combined using the multiplication fusion method, offer an effective approach for VQA tasks. These insights contribute to the field of multimodal deep learning by identifying optimal model configurations and fusion techniques that enhance VQA performance in both synthetic and real-world scenarios.

Perspectives and Future Work

The potential applications of our findings extend significantly, especially in the medical domain. Integrating our VQA models with medical imaging can revolutionize predictive diagnostics and patient monitoring. By combining medical images with patient records and clinical notes, these models can provide comprehensive assessments and predictions, potentially improving diagnostic accuracy and patient outcomes.

Moreover, the synergy between computer vision and natural language processing in VQA highlights the potential for incorporating additional modalities, such as audio or sensor data. This could enhance AI systems' capabilities in tackling complex problems in areas like autonomous driving, robotics, and smart home technologies.

Future research should address challenges in multimodal representation, translation, alignment, fusion, and co-learning to develop even more robust models. These advancements will likely lead to improved performance across a wider range of tasks, broadening the applications in healthcare, education, security, and beyond.

References

Acheampong, F. A., Nunoo-Mensah, H., & Chen, W. (2021). Transformer models for text-based emotion detection: A review of BERT-based approaches. *Artificial Intelligence Review*, *54*(8), 5789–5829. https://doi.org/10.1007/s10462-021-09958-2

Alzubaidi, L., Zhang, J., Humaidi, A. J., Al-Dujaili, A., Duan, Y., Al-Shamma, O., Santamaría, J., Fadhel, M. A., Al-Amidie, M., & Farhan, L. (2021). Review of deep learning: Concepts, CNN architectures, challenges, applications, future directions. *Journal of Big Data*, *8*(1), 53. https://doi.org/10.1186/s40537-021-00444-8

Arulkumaran, K., Deisenroth, M. P., Brundage, M., & Bharath, A. A. (2017). Deep reinforcement learning: A brief survey. *IEEE Signal Processing Magazine*, *34*(6), 26–38.

Atrey, P. K., Hossain, M. A., El Saddik, A., & Kankanhalli, M. S. (2010). Multimodal fusion for multimedia analysis: A survey. *Multimedia Systems*, *16*(6), 345–379. https://doi.org/10.1007/s00530-010-0182-0

Baltrušaitis, T., Ahuja, C., & Morency, L.-P. (2017). *Multimodal Machine Learning: A Survey and Taxonomy* (arXiv:1705.09406). arXiv. http://arxiv.org/abs/1705.09406

Barua, A., Ahmed, M. U., & Begum, S. (2023). A Systematic Literature Review on Multimodal Machine Learning: Applications, Challenges, Gaps and Future Directions. *IEEE Access*, *11*, 14804–14831. https://doi.org/10.1109/ACCESS.2023.3243854

Boulahia, S. Y., Amamra, A., Madi, M. R., & Daikh, S. (2021). Early, intermediate and late fusion strategies for robust deep learning-based multimodal action recognition. *Machine Vision and Applications*, *32*(6), 121. https://doi.org/10.1007/s00138-021-01249-8

Crawford, M., Khoshgoftaar, T. M., Prusa, J. D., Richter, A. N., & Al Najada, H. (2015). Survey of review spam detection using machine learning techniques. *Journal of Big Data*, *2*(1), 23.

https://doi.org/10.1186/s40537-015-0029-9

Cunningham, P., Cord, M., & Delany, S. J. (2008). Supervised Learning. In M. Cord & P. Cunningham (Eds.), *Machine Learning Techniques for Multimedia* (pp. 21–49). Springer Berlin Heidelberg. https://doi.org/10.1007/978-3-540-75171-7_2

Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., Uszkoreit, J., & Houlsby, N. (2021). *An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale* (arXiv:2010.11929). arXiv. http://arxiv.org/abs/2010.11929

Elliott, D., & Kádár, Á. (2017). *Imagination improves Multimodal Translation* (arXiv:1705.04350). arXiv. http://arxiv.org/abs/1705.04350

Elman, J. L. (1990). Finding Structure in Time. *Cognitive Science*, *14*(2), 179–211. https://doi.org/10.1207/s15516709cog1402_1

Hoang, V.-T., & Jo, K.-H. (2021). Practical Analysis on Architecture of EfficientNet. *2021 14th International Conference on Human System Interaction (HSI)*, 1–4. https://doi.org/10.1109/HSI52170.2021.9538782

Jabeen, S., Li, X., Amin, M. S., Bourahla, O., Li, S., & Jabbar, A. (2023). A Review on Methods and Applications in Multimodal Deep Learning. *ACM Transactions on Multimedia Computing, Communications, and Applications*, *19*(2s), 1–41. https://doi.org/10.1145/3545572

Janiesch, C., Zschech, P., & Heinrich, K. (2021). Machine learning and deep learning. *Electronic Markets*, *31*(3), 685–695. https://doi.org/10.1007/s12525-021-00475-2

Johnson, J., Hariharan, B., van der Maaten, L., Fei-Fei, L., Zitnick, C. L., & Girshick, R. (2016). *CLEVR: A Diagnostic Dataset for Compositional Language and Elementary Visual Reasoning* (arXiv:1612.06890). arXiv. http://arxiv.org/abs/1612.06890

Kafle, K., & Kanan, C. (2017). Visual Question Answering: Datasets, Algorithms, and Future Challenges. *Computer Vision and Image Understanding*, *163*, 3–20. https://doi.org/10.1016/j.cviu.2017.06.005

Kenton, J. D. M.-W. C., & Toutanova, L. K. (2019). Bert: Pre-training of deep bidirectional transformers for language understanding. *Proceedings of naacL-HLT*, *1*, 2. https://aclanthology.org/N19-1423.pdf?utm_medium=ema

Lau, J. J., Gayen, S., Ben Abacha, A., & Demner-Fushman, D. (2018). A dataset of clinically generated visual questions and answers about radiology images. *Scientific Data*, *5*(1), 180251. https://doi.org/10.1038/sdata.2018.251

LeCun, Y., Bengio, Y., & Hinton, G. (2015). Deep learning. *Nature*, *521*(7553), 436–444. https://doi.org/10.1038/nature14539

Lu, D., & Weng, Q. (2007). A survey of image classification methods and techniques for improving classification performance. *International Journal of Remote Sensing*, *28*(5), 823–870. https://doi.org/10.1080/01431160600746456

Marois, V., Jayram, T. S., Albouy, V., Kornuta, T., Bouhadjar, Y., & Ozcan, A. S. (2018). *On transfer learning using a MAC model variant* (arXiv:1811.06529). arXiv. http://arxiv.org/abs/1811.06529

McCulloch, W. S., & Pitts, W. (1943). A logical calculus of the ideas immanent in nervous activity. *The Bulletin of Mathematical Biophysics*, *5*(4), 115–133. https://doi.org/10.1007/BF02478259

Patwardhan, N., Marrone, S., & Sansone, C. (2023). Transformers in the real world: A survey on nlp applications. *Information*, *14*(4), 242.

Rahate, A., Walambe, R., Ramanna, S., & Kotecha, K. (2022). Multimodal Co-learning: Challenges, applications with datasets, recent advances and future directions. *Information Fusion*, *81*, 203–239. https://doi.org/10.1016/j.inffus.2021.12.003

Rasenberg, M., Özyürek, A., & Dingemanse, M. (2020). Alignment in Multimodal Interaction: An Integrative Framework. *Cognitive Science*, *44*(11), e12911. https://doi.org/10.1111/cogs.12911

Russell, S. J., & Norvig, P. (2016). *Artificial intelligence: A modern approach*. Pearson. https://thuvienso.hoasen.edu.vn/handle/123456789/8967

Sharma, S., Sharma, S., & Athaiya, A. (2017). Activation functions in neural networks. *Towards Data Sci*, *6*(12), 310–316.

Smagulova, K., & James, A. P. (2019). A survey on LSTM memristive neural network architectures and applications. *The European Physical Journal Special Topics*, *228*(10), 2313–2324. https://doi.org/10.1140/epjst/e2019-900046-x

Sorzano, C. O. S., Vargas, J., & Montano, A. P. (2014). *A survey of dimensionality reduction techniques* (arXiv:1403.2877). arXiv. http://arxiv.org/abs/1403.2877

Sykes, A. O. (1993). *An introduction to regression analysis*. https://chicagounbound.uchicago.edu/cgi/viewcontent.cgi?article=1050&context=law_and_economics

Tan, M., & Le, Q. (2019). Efficientnet: Rethinking model scaling for convolutional neural networks. *International Conference on Machine Learning*, 6105–6114. http://proceedings.mlr.press/v97/tan19a.html?ref=jina-ai-gmbh.ghost.io

Targ, S., Almeida, D., & Lyman, K. (2016). *Resnet in Resnet: Generalizing Residual Architectures* (arXiv:1603.08029). arXiv. http://arxiv.org/abs/1603.08029

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., & Polosukhin, I. (2017). Attention is all you need. *Advances in Neural Information Processing Systems*, *30*. https://proceedings.neurips.cc/paper/7181-attention-is-all

Vedaldi, A., & Zisserman, A. (2016). Vgg convolutional neural networks practical. *Department of Engineering Science, University of Oxford*, *66*. https://www.academia.edu/download/49291782/CNN_practical.pdf

Wu, Q., Teney, D., Wang, P., Shen, C., Dick, A., & Hengel, A. van den. (2016). *Visual Question Answering: A Survey of Methods and Datasets* (arXiv:1607.05910). arXiv. http://arxiv.org/abs/1607.05910

Yue, S., Li, P., & Hao, P. (2003). SVM classification:Its contents and challenges. *Applied Mathematics-A Journal of Chinese Universities*, *18*(3), 332–342. https://doi.org/10.1007/s11766-003-0059-5

Zhou, D., Kang, B., Jin, X., Yang, L., Lian, X., Jiang, Z., Hou, Q., & Feng, J. (2021). *DeepViT: Towards Deeper Vision Transformer* (arXiv:2103.11886). arXiv. http://arxiv.org/abs/2103.11886

Webographie

Aramendia, A. I. (2024, January 16). Convolutional Neural Networks: A Complete Guide. Medium. https://medium.com/@alejandro.itoaramendia/convolutional-neural-networks-cnns-a-complete-guide-a803534a1930 (visited on 15/06/2024)

Balaji, S. (2023, August 26). Binary Image Classifier CNN using TensorFlow. Medium. https://medium.com/techiepedia/binary-image-classifier-cnn-using-tensorflow-a3f5d6746697 (visited on 10/06/2024)

Bangar, S. (2022, July 5). Resnet Architecture Explained. Medium. https://medium.com/@siddheshb008/resnet-architecture-explained-47309ea9283d (visited on 20/06/2024)

Boesch, G. (2024, June 20). Very Deep Convolutional Networks (VGG) Essential Guide. viso.ai. https://viso.ai/deep-learning/vgg-very-deep-convolutional-networks/ (visited on 23/06/2024)

Franky. (2021, October 6). Deep Learning and Visual Question Answering. Medium. https://towardsdatascience.com/deep-learning-and-visual-question-answering-c8c8093941bc (visited on 22/06/2024)

Qayyum, R. (2022, August 16). Introduction to Pooling Layers in CNN. Medium. https://pub.towardsai.net/introduction-to-pooling-layers-in-cnn-dafe61eabe34 (visited on 13/06/2024)

Singh, G. (2024, June 4). Understanding Architecture of LSTM. Analytics Vidhya. https://www.analyticsvidhya.com/blog/2021/01/understanding-architecture-of-lstm/ (visited on 18/06/2024)

Structure of joint and coordinated representations. (n.d.). ResearchGate. https://www.researchgate.net/figure/Structure-of-joint-and-coordinated-representations_fig5_341205866 (visited on 18/06/2024)

The main types of machine learning. Main approaches include... (n.d.). ResearchGate. https://www.researchgate.net/figure/The-main-types-of-machine-learning-Main-approaches-include-classification-and_fig1_354960266 (visited on 09/06/2024)

The Transformer based BERT base architecture with twelve... (n.d.). ResearchGate. https://www.researchgate.net/figure/The-Transformer-based-BERT-base-architecture-with-twelve-encoder-blocks_fig2_349546860 (visited on 18/06/2024)

This figure shows three different activation functions... (n.d.). ResearchGate. https://www.researchgate.net/figure/This-figure-shows-three-different-activation-functions-graphs-and-its-equations-From_fig4_322537764 (visited on 09/06/2024)

Vinija's Notes • Multimodal machine learning • Multimodal challenges. (n.d.). https://vinija.ai/multimodal/challenges/#google_vignette (visited on 20/06/2024)

Vision Transformer (ViT) model architecture. (n.d.). ResearchGate. https://www.researchgate.net/figure/Vision-Transformer-ViT-model-architecture-ViT-breaks-the-image-into-patches-and-treats_fig2_362567384 (visited on 18/06/2024)

Abstract

This thesis investigates the field of Visual Question Answering (VQA), an advanced application of multimodal deep learning that combines computer vision and natural language processing to enable machines to answer questions about visual content. The primary objective is to develop a robust VQA model capable of interpreting and reasoning about complex visual scenes in a human-like manner. Utilizing various models VGG19, Resnet50, efficientnet_b5 and Vit for image feature extraction and a BERT-based, lstm model for question feature extraction, the study employs a mid-level fusion method through element-wise multiplication or concatenation to combine these features.

The research leverages datasets such as CLEVR, which is designed to test true reasoning abilities by providing 3D-rendered objects with detailed questions, and VQA-RAD for real world problems, which supports clinical decision-making through radiology images paired with relevant questions and answers.

Experimental results showed that the combination of ViT, BERT, and the multiplication model performed exceptionally well, achieving high accuracies of 98.70% on the CLEVR dataset and 96% on the VQA-RAD dataset. This made it the most effective model for both general and medical visual question answering.

Keywords: Visual Question Answering, Multimodal deep learning, Computer vision, Natural language processing, Mid-level fusion.

Résumé

Cette thèse explore le domaine de la réponse visuelle aux questions (VQA), une application avancée de l'apprentissage profond multimodal qui combine la vision par ordinateur et le traitement du langage naturel pour permettre aux machines de répondre à des questions sur le contenu visuel. L'objectif principal est de développer un modèle VQA robuste capable d'interpréter et de raisonner sur des scènes visuelles complexes de manière similaire à celle des humains. En utilisant divers modèles tels que VGG19, ResNet50, EfficientNet-B5 et ViT pour l'extraction de caractéristiques d'image, et un modèle basé sur BERT ou LSTM pour l'extraction de caractéristiques des questions, l'étude emploie une méthode de fusion de niveau intermédiaire par multiplication élémentaire ou concaténation pour combiner ces caractéristiques.

La recherche s'appuie sur des jeux de données comme CLEVR, conçu pour tester les véritables capacités de raisonnement en fournissant des objets rendus en 3D avec des questions détaillées, et VQA-RAD, destiné à des problèmes réels, qui soutient la prise de décision clinique à travers des images de radiologie accompagnées de questions et réponses pertinentes.

Les résultats expérimentaux ont montré que la combinaison du modèle ViT, BERT et la technique de multiplication a excellé, atteignant des précisions élevées de 98,70 % sur le jeu de données CLEVR et de 96 % sur le jeu de données VQA-RAD, en faisant le modèle le plus efficace pour la réponse aux questions visuelles générales et médicales.

Mots clé : Réponse visuelle aux questions, Apprentissage profond multimodal, Vision par ordinateur, Traitement du langage naturel, Fusion de niveau intermédiaire.

ملخص:

تبحث هذه الأطروحة في مجال الإجابة على الأسئلة البصرية (VQA)، وهو تطبيق متقدم للتعلم العميق متعدد الوسائط يجمع بين رؤية الكمبيوتر ومعالجة اللغة الطبيعية لتمكين الآلات من الإجابة على الأسئلة المتعلقة بالمحتوى البصري. الهدف الرئيسي هو تطوير نموذج VQA قوي قادر على تفسير المشاهد البصرية المعقدة والتفكير فيها بطريقة تشبه البشر. باستخدام نماذج متنوعة مثل 19VGG و 50ResNet و 5EfficientNet-B و ViT لاستخراج ميزات الصور، ونموذج يعتمد على BERT أو LSTM لاستخراج ميزات الأسئلة، تعتمد الدراسة على طريقة دمج في المستوى المتوسط من خلال الضرب العنصري أو الدمج لتوحيد هذه الميزات.

تستند الأبحاث إلى مجموعات بيانات مثل CLEVR، المصممة لاختبار القدرات الحقيقية في الاستدلال من خلال تقديم كائنات ثلاثية الأبعاد مع أسئلة مفصلة، و VQA-RAD التي تدعم اتخاذ القرارات السريرية من خلال صور الأشعة المقرونة بأسئلة وإجابات ذات صلة.

أظهرت النتائج التجريبية أن الجمع بين نموذج ViT وBERT وتقنية الضرب حقق أداءً متميزًا، حيث بلغت دقته 98.70٪ على مجموعة بيانات CLEVR و96٪ على مجموعة بيانات VQA-RAD، مما يجعله النموذج الأكثر فعالية للإجابة عن الأسئلة البصرية العامة والطبية.

الكلمات المفتاحية: الإجابة على الأسئلة البصرية، التعلم العميق متعدد الوسائط، رؤية الكمبيوتر، معالجة اللغة الطبيعية، دمج في المستوى المتوسط.