Arabic Sign Language Comparative Analysis

(Project Report)

Authors:

Kamel Mojtaba, Amin Alawad,, Obai Ali, and Ahmed Alnasri
Multimedia University, 1191301456@student.mmu.edu.my, 1191302190@student.mmu.edu.my, 1171103208@student.mmu.edu.my, 1211300174@student.mmu.edu.my.

Abstract **- There are 2 types of people who cannot communicate in the same way as others. Deaf and hard-of-hearing people, both of them use sign language for their communication with other people. Sign language includes different types of hand gestures and facial expressions for communication and emotional expression so this paper focuses on the development and implementation of an innovative Arabic Sign Language (ASL) recognition system designed to bridge communication gaps for the deaf and hard-of-hearing community. Utilizing advanced deep learning models, including Convolutional Neural Networks (CNNs), Long Short-Term Memory (LSTM), and Transfer Learning with pre-trained models, this research aims to identify the most effective model(s) based on accuracy, efficiency, and practical usability. This paper will detail the methodology employed in evaluating the models, including performance metrics such as accuracy, precision, recall, and F1-score, loss, alongside qualitative visualizations**

Index Terms - Arabic Sign Language, Comparative Analysis, Deep Learning, Gesture Recognition, Accessibility, Assistive Technologies..

Introduction

The study of Arabic Sign Language (ArSL) recognition has seen significant advancements in recent years, propelled by the integration of deep learning techniques that have substantially improved the accuracy and efficiency of sign language interpretation systems(Othman & El Ghoul, 2021). This project aims to conduct a comprehensive comparative analysis of Arabic Sign Language recognition by employing a variety of machine learning models, including Convolutional Neural Networks (CNNs), Long Short-Term Memory (LSTM) networks, and Transfer Learning with pre-trained models. These models represent the forefront of research in both the fields of machine learning and sign language recognition, offering unique strengths in feature selection, feature extraction, and model training processes.

Recent studies have highlighted the effectiveness of CNNs in processing spatial features from visual data, making them particularly suited for sign language recognition (Papatsimouli et al., 2022). Similarly, LSTM networks have been recognized for their capacity to handle sequential data, capturing the temporal dynamics of sign languageAbu-Jamie & Abu-Naser, 2022). Transfer Learning, utilizing pre-trained models, has been identified as a powerful method to leverage existing knowledge, significantly reducing the need for extensive data collection specific to Arabic Sign Language.

This project will not only undertake the task of implementing these models for Arabic Sign Language recognition but will also perform a comparative analysis to evaluate their performance comprehensively. The comparative analysis aims to elucidate the strengths and weaknesses of each model in the context of Arabic Sign Language, providing insights into the most effective strategies for future research and application development in this area.

Through meticulous literature review, feature selection, feature extraction, model training, and comparative analysis, this project endeavors to contribute to the burgeoning field of Arabic Sign Language recognition, paving the way for more accessible and effective communication tools for the Arab deaf community.

Literature Review

Sign languages require many processes, such as hand configuration recognition, motion discrimination, identification of facial expressions, and recognition of linguistically relevant spatial contrasts (Papatsimouli et al., 2022). Several studies have gone into sign language recognition and translation. In this paper, we focus on exploring the effectiveness of some models like LSTM (Long Short-Term Memory), CNN (Convolutional Neural Network), and Graph Neural Networks (GNNs) architectures to enhance translation accuracy,
A lot of researchers used LSTM models in recognizing static sign language gestures for example (Boondamnoen, Thongsri, Sahabantoegnsin, & Woraratpanya, 2023) focuses on Indian Sign Language (ISL) gestures and illustrates the potential of the Media pipe Holistic framework in

conjunction with LSTM models, leading to heightened accuracy in gesture recognition. Additionally, (Yin, 2020) offers a comprehensive understanding of automatic sign language recognition, highlighting the significance of including both hand and non-manual features. This comprehensive view aids our experimental framework and subsequent LSTM CNN comparison.

Meanwhile, (Pandian et al., 2023) delves into the CNN model specialized in recognizing American Sign Language, adding depth to the exploration. Moreover, (Sharma & Singh, 2021) delves into the domain of deep CNN models for sign language translation, contributing its strengths to the field. The utilization of CNN architectures in sign language recognition and translation is another area of significant strength and exploration. Notably, (Papatsimouli et al., 2022) stands out for its comparative analysis of LSTM and Transformer architectures within the realm of sign language translation, contributing valuable insights into their respective strengths.

ResNet models, specifically ResNet and 3D-ResNet architectures, present a promising approach to improving recognition accuracy. These models leverage deep neural networks with residual connections, enabling the learning of complex patterns in sign language gestures through both spatial and temporal dimensions. The ResNet architecture, known for mitigating the vanishing gradient problem and allowing for deeper network structures, is particularly advantageous in capturing the intricate details of hand gestures and facial expressions essential for sign language interpretation.

Recent studies, such as those by Tanseem N. Abu-Jamie and Prof. Dr. Samy S. Abu-Naser, and Shiqi Wang et al (Abu-Jamie & Abu-Naser, 2022)., have demonstrated the effectiveness of ResNet and improved 3D-ResNet algorithms in sign language recognition tasks. These approaches have achieved significant accuracy improvements by focusing on enhanced feature extraction methods.

including attention to hand features and the integration of motion and spatial information, showcasing the potential of ResNet models in advancing sign language recognition technologies.

Methodology

Our image classification methodology integrates advanced preprocessing techniques with robust deep learning models to facilitate accurate predictions. The process begins with a comprehensive database and culminates in a detailed evaluation of the model's predictive performance. The methodology follows a systematic approach with clearly defined stages, as outlined in the following steps.

Preprocessing:

- YOLO Model: Initially, the images from the database are processed using a YOLO (You Only Look Once) model. This step is critical where the model identifies and classifies objects within the images. The output is then used for feature extraction.
- Feature Extraction: Following the YOLO model's detection phase, pertinent features are extracted from the images. These features serve as the foundational data for training the classification models, encompassing vital visual cues necessary for accurate classification.

Dataset Partitioning

After preprocessing, the data is divided into three distinct sets:

- Training Dataset: This subset is used to train the model, allowing the algorithm to learn from the data's features and make generalizations about the images it contains.
- Validation Dataset: Separate from the training data, the validation dataset is utilized to fine-tune model parameters and prevent overfitting, ensuring the model's ability to generalize well to new, unseen data.
- Testing Dataset: This dataset is not used during the training phase. It serves as a final, unbiased evaluation of the model's classification performance.

Model Training and Validation

- Model Training: The models are trained on the training dataset, where they iteratively learn to classify images correctly. This phase involves adjusting the weights of the network based on the error rates in classification, using optimization algorithms to minimize these errors.
- Model Validation: Concurrently, the models are validated using the validation dataset. This process helps in monitoring the models' performance on non-training data, optimizing hyperparameters, and selecting the best-performing model.

Prediction and Evaluation

- Prediction: Once the models are trained and validated, they are used to predict the classes of images in the testing dataset. These predictions are based on the learned features and patterns during the training phase.
- Evaluation: The final step involves evaluating the models' predictions to determine their accuracy and effectiveness. Standard metrics such as accuracy, precision, recall, and F1 score are calculated to provide a comprehensive assessment of each model's performance.

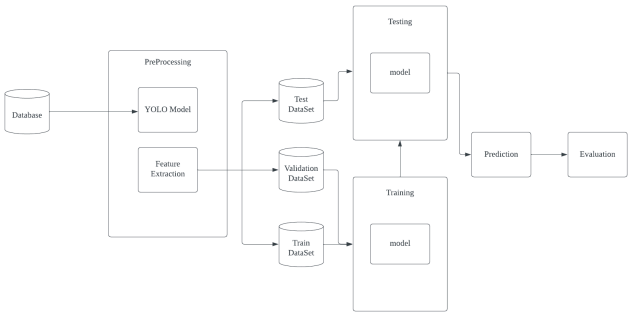

FIGURE 1
METHODOLOGY OVERFLOW

PREPROCESSING

The image preprocessing phase is critical in preparing the input data for subsequent analysis and classification. This phase consists of a series of operations, starting with image acquisition, followed by the detection and isolation of the hand within the image. The initial step involves resizing the acquired images to a uniform dimension of 64x64 pixels. This standardization is essential for ensuring consistency across the dataset, facilitating more efficient processing and analysis by the subsequent models.

Upon resizing, the RGB images are then subjected to the YOLOv5 model, which is tasked with the detection of the hand region within the image. The YOLO (You Only Look Once) model. By employing the YOLOv5 model, the system can accurately identify and localize the hand within the image frame, irrespective of the hand's position, orientation, or the presence of complex backgrounds.

Following the successful detection of the hand by the YOLOv5 model, the next step involves cropping the image to the boundaries of the detected hand region. This cropping operation is pivotal in generating a clean image that contains only the hand. By doing so, it significantly reduces noise and eliminates background elements that are irrelevant to the sign language classification task. This preprocessing step is instrumental in enhancing the focus on the gesture being made, thereby improving the accuracy of the subsequent classification models.

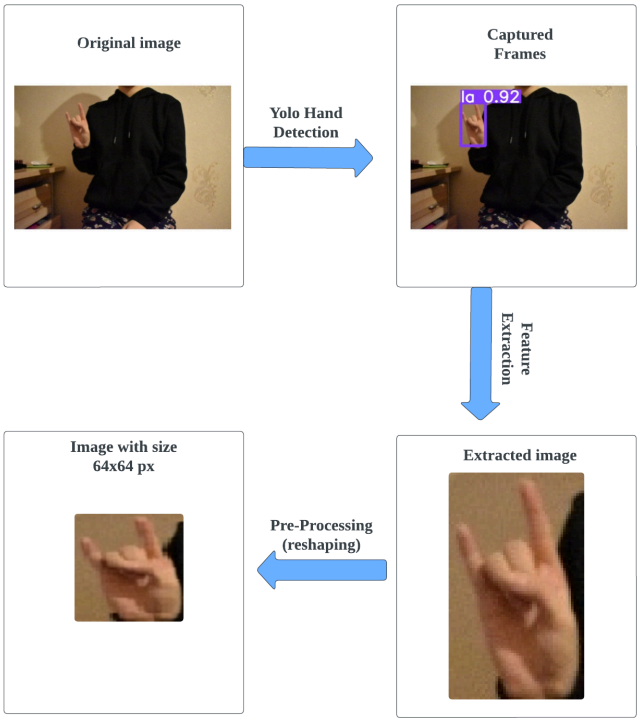

FIGURE 2
DATA PRE-PROCESSING

MODELS IMPLEMENTATION

CNN:

Convolutional Neural Network (CNN) tailored for image classification tasks, particularly effective in recognizing complex visual patterns such as those found in Arabic Sign Language. It employs a layered approach, starting with convolutional blocks that progressively increase in complexity and depth, from 32 filters in the initial layer up to 256 filters in the final convolutional block. Each block consists of a convolutional layer with ReLU activation for non-linearity, followed by batch normalization to ensure stable learning, and max pooling to reduce spatial dimensions, thereby focusing on the most significant features while minimizing computational requirements. Dropout layers are strategically placed to mitigate overfitting by randomly omitting a fraction of the neurons during training, enhancing the model's ability to generalize across unseen data.

The architecture culminates in a flattening step that transforms the multidimensional feature maps into a vector, facilitating the transition to dense layers. A substantial dense layer with 512 neurons follows, incorporating ReLU activation to interpret the features extracted by the convolutional stages. Another dropout layer is introduced before the final classification layer to further regularize the model. The concluding layer employs a softmax activation function, which outputs a

probability distribution over the various classes, each corresponding to a distinct sign in the Arabic Sign Language. This structure is adept at extracting and interpreting high-level features from images, making it highly suitable for applications requiring precise pattern recognition and classification, such as sign language interpretation.

LSTM:

LSTM (Long Short-Term Memory) network is crafted for processing sequences, making it adept at handling tasks where understanding temporal dynamics is key. It starts by reshaping the input to align with LSTM's requirements, transforming image data into a sequence format by flattening dimensions except for the width. This adjustment allows the network to treat the image frames as sequential data, focusing on the temporal progression of features.

The core of the model consists of two LSTM layers: the first with 64 units processes the sequence in full, ensuring the temporal information is captured in each step. It passes this detailed sequence to the second LSTM layer with 32 units, which then condenses the temporal data into a more abstract representation. This hierarchical processing allows the model to capture and interpret complex temporal patterns effectively.

To combat overfitting, a dropout layer with a 50% rate is introduced after the LSTM layers, randomly disabling neurons during training to promote a more generalized model that performs well on unseen data. The model transitions to a dense layer with 64 units and ReLU activation, serving as a bridge to the final classification stage.

The concluding layer is a softmax-activated dense layer, translating the LSTM's abstract temporal understanding into a probability distribution across the predefined classes. This setup makes the model particularly suitable for tasks like gesture recognition in videos, where the sequence and timing of movements are crucial for accurate classification. The LSTM model's ability to parse and learn from temporal sequences offers a powerful tool for applications requiring nuanced understanding of time-based patterns.

ResNet:

This model leverages a pre-trained ResNet architecture as its base, enhancing it with additional layers to tailor the network for specific classification tasks. ResNet, known for its deep architecture and residual connections that help mitigate the vanishing gradient problem, provides a robust foundation for feature extraction. The model builds upon this with a sequence of layers designed to refine and adapt these features for precise classification.

After the ResNet base, a Global Average Pooling 2D layer follows, which reduces the spatial dimensions of the feature maps to a single vector per map, effectively summarizing the most critical features while minimizing overfitting risks and reducing the model's complexity.

The next stages of the model include dense layers with ReLU activation to introduce non-linearity and enhance the model's capacity to learn complex patterns. The first dense layer has 256 units, followed by batch normalization to maintain stability during training by normalizing the layer's inputs. A dropout layer with a rate of 0.3 follows, reducing overfitting by randomly omitting a portion of the neurons, forcing the network to learn more robust features. This pattern is repeated with another dense layer of 128 units, accompanied by batch normalization and dropout, further refining the model's ability to classify accurately.

The final layer is a dense layer with a softmax activation function, designed to output a probability distribution over the classes, indicating the model's prediction for the input image. This architecture effectively combines the powerful feature extraction capabilities of ResNet with additional layers that tailor the network's output for specific classification tasks, making it highly effective for a wide range of applications, from image recognition to more specialized tasks that benefit from deep, nuanced feature understanding

TRAINING AND VALIDATION

CNN validation accuracy:

The graph depicts the training and validation accuracy of a Convolutional Neural Network (CNN) model over approximately 50 epochs. The training accuracy increases over time, showing that the model is learning from the training data. The training accuracy surpasses 90%, suggesting that the model fits the training data well. Also, there is a rapid increase in the validation accuracy initially, which then fluctuates but generally maintains an upward trend. The fluctuations might be due to the model learning new patterns that aren't necessarily generalizable or to the inherent noise within the validation dataset.

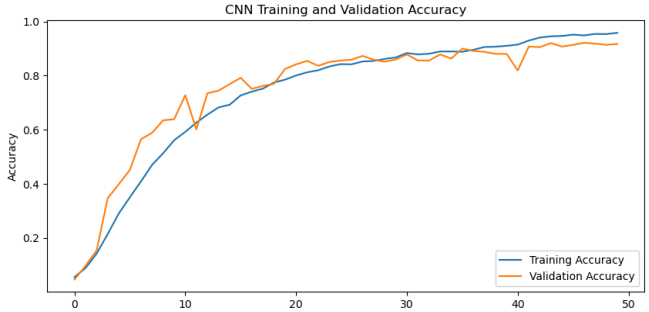

FIGURE 3
CNN VALIDATION ACCURACY

LSTM validation accuracy:

The graph illustrates the training and validation accuracy of an LSTM model across approximately 50 epochs.
Both training and validation accuracies increase gradually over the epochs. Unlike typical accuracy curves, which might show a steep climb, this graph indicates a more moderate learning curve. This can be characteristic of complex sequence learning tasks, where LSTM networks are often applied.
You should note that the highest accuracy reached is around 40% for training and slightly less for validation. This is relatively low for most machine learning tasks, suggesting a challenging problem domain or that the model may require further tuning and training.

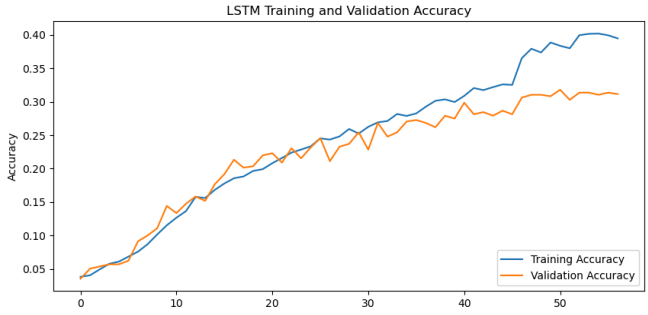

FIGURE 4
LSTM VALIDATION ACCURACY

RustNet validation accuracy:

In this model, both the training and validation accuracy increase sharply during the initial epochs, which suggests that the model is learning quickly from the data. but As the epochs increase, both accuracies converge, with the validation accuracy slightly below the training accuracy. This indicates a good generalization of the model on unseen data. However, the fact that they are close suggests that the model isn't overfitting, which is a positive sign of a well-tuned model.

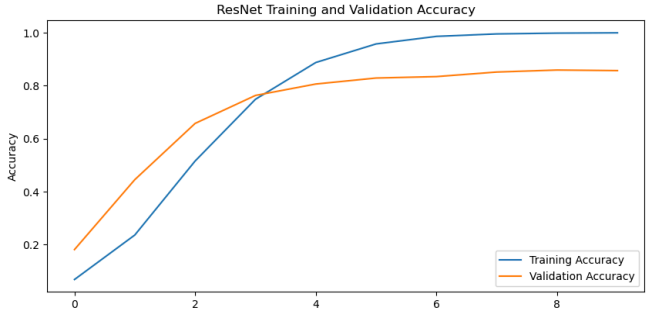

FIGURE 5
RESTNET VALIDATION ACCURACY

CNN validation Loss:

There is a sharp decrease in both training and validation losses at the beginning, typical of the rapid learning phase of CNNs.there are fluctuations in the validation loss, which may suggest that the model's learning is less stable. These could be due to the variance in the validation data or the model's sensitivity to certain features.The training loss continues to decrease and plateau, while the validation loss plateaus with minor fluctuations. This indicates that further training may not yield significant improvements in learning.

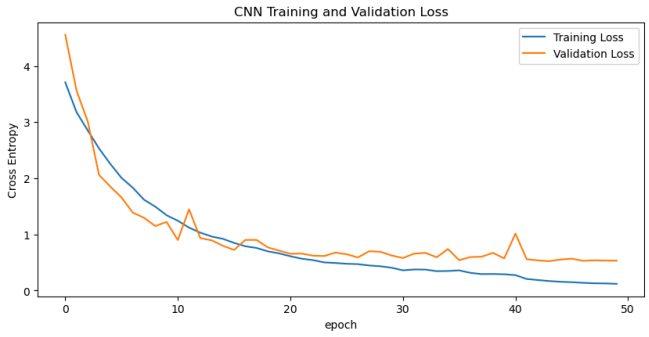

FIGURE 6
CNN VALIDATION LOSS

LTSM validation Loss:

Both training and validation losses decrease over time, which indicates that the LSTM model is learning from the training data.The training and validation losses converge as epochs increase, which is a positive sign that the model is generalizing well and not overfitting significantly.

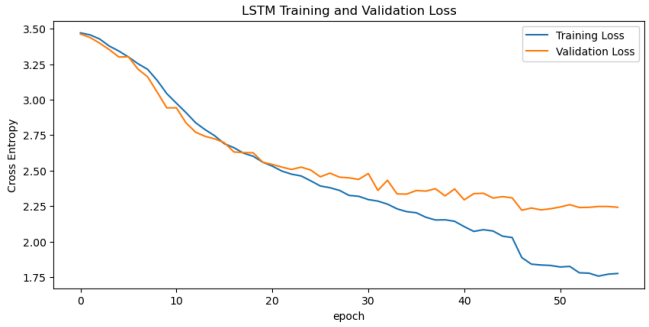

FIGURE 7
LSTM VALIDATION LOSS

RustNet validation Loss:

The graph shows a steep decline in loss for both training and validation, which is characteristic of the quick feature learning capability of ResNet.The validation loss is very close to the training loss, and both are low, indicating good model performance and generalization from early on in training.

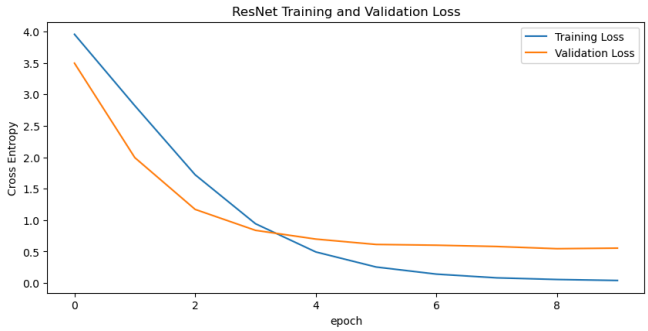

FIGURE 8
REST VALIDATION LOSS

EVALUATION

Evaluation Matrix

ResNet Performance:

Loss: 0.3955
Accuracy: 89.65%
ResNet's performance suggests a relatively low loss and a high accuracy rate. The model seems to generalize well to the validation data, indicating a well-fitted model that is likely to perform effectively on unseen data.

CNN Performance:

Loss: 0.3983
Accuracy: 93.21%
The CNN model exhibits a loss comparable to ResNet but boasts a higher accuracy. This indicates that the CNN model has learned the features of the dataset effectively and suggests a strong ability to recognize Arabic Sign Language signs accurately.

LSTM Performance:

Loss: 2.1655
Accuracy: 34.59%
The LSTM model shows a significantly higher loss and lower accuracy compared to the ResNet and CNN models. This might be due to the complexity of the sequence prediction in the sign language recognition task or might indicate that the LSTM model requires further optimization. The lower performance could also suggest that the LSTM model may struggle with the dataset or task compared to the other models.

TABLE 1
MODEL EVALUATION COMPARATIVE

Model	Accuracy	Precession	recall	F1 score
ResNet	0.8966	0.8973	0.8940	0.8933
CNN	0.9321	0.9335	0.9333	0.9319
LSTM	0.3459	0.3415	0.3451	0.3311

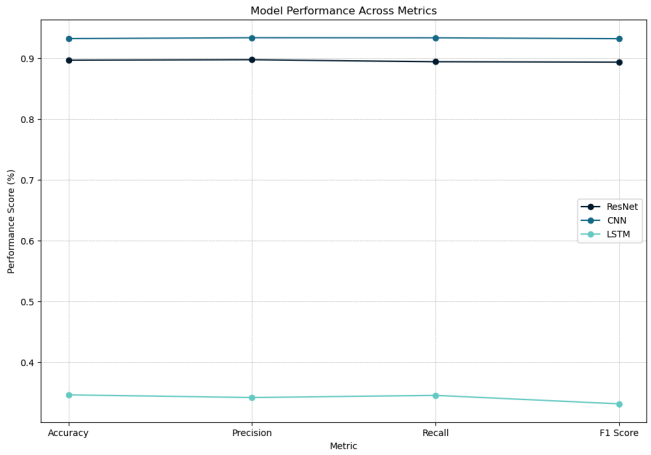

FIGURE 9
MODEL PERFORMANCE MATRIX

From the graph, we can see that ResNet and CNN Both models have identical performance across all four metrics, according to the graph. They score very high on accuracy, precision, recall, and F1 score, consistently close to or at 90%. This suggests that these models are not only good at correctly identifying the correct signs (high accuracy) but are also precise (high precision), generate few false negatives (high recall), and balance precision and recall effectively (high F1 score). On the other hand, The LSTM model has significantly lower scores in all metrics, particularly accuracy and F1 score. It is markedly below the other two models, which indicates that it is less effective at this particular task. The low accuracy and F1 score suggest that it has a higher rate of both false positives and false negatives.

So based on our results, we can say that CNN and Resnet are better option when dealing with sign language

Challenges and limitations:

Convolutional Neural Networks (CNNs):

- Overfitting: CNNs are highly prone to overfitting, particularly when trained on limited datasets. Techniques like dropout and data augmentation can mitigate this, but ensuring generalization remains a significant challenge.
- Computational Intensity: The training of CNNs, especially with deep architectures, demands substantial computational resources, leading to increased training times and necessitating GPUs or distributed computing for practical training periods.
- Interpretability Issues: CNNs, characteristic of many deep learning models, lack interpretability. This opacity makes it difficult to understand their decision-making processes, posing a challenge in applications requiring transparency.

Residual Networks (ResNets):

- Complexity and Resource Demands: ResNets, designed to enable deeper networks through skip connections, increase computational resource demands, which can be a limiting factor for resource-constrained projects.
- Optimization Challenges: Despite facilitating the training of deeper networks, ResNets can present optimization challenges, such as slow convergence and difficulty in hyperparameter tuning, particularly as network depth increases.
- Degradation Phenomenon: ResNets aim to avoid performance degradation with increased depth; however, consistently improving performance with added depth requires meticulous architecture planning and tuning.

Long Short-Term Memory Networks (LSTMs):

- Training and Computational Resources: LSTMs require extensive computational resources due to their complex architecture, leading to prolonged training times compared to simpler models.
- Vanishing and Exploding Gradients: Designed to mitigate vanishing gradients, LSTMs can still face both vanishing and exploding gradient issues, complicating training stability.
- Parallelization Challenges: The sequential dependency of LSTM operations hinders parallelization, affecting training efficiency compared to models like CNNs, where operations can be efficiently parallelized.
- Model Complexity and Overfitting: The complexity of LSTMs increases the risk of overfitting, especially with smaller datasets, necessitating regularization techniques such as dropout which can complicate training.
- Long-Term Dependency Limitations: Despite their design, LSTMs may struggle with very long sequences or complex temporal patterns, impacting their effectiveness in capturing extended temporal dynamics.

Conclusion

This report has undertaken a comprehensive comparative analysis of various deep learning models, specifically CNNs, ResNets, and LSTMs, in addressing the complex problem of image classification. Each model, with its unique architecture and capabilities, has demonstrated particular strengths in capturing the intricacies of visual data, offering insights into the evolving landscape of machine learning techniques for computer vision.

CNNs, with their profound ability to capture spatial hierarchies in images, have underscored the importance of deep learning in extracting meaningful patterns from visual data. ResNets, through their innovative use of skip connections, have pushed the boundaries of model depth, enabling the training of significantly deeper networks without succumbing to the vanishing gradient problem. LSTMs have introduced a novel approach to handling temporal and sequential information, although less commonly applied to image classification directly, they underscore the versatility and adaptability of neural networks in processing complex data structures.

However, this analysis has also highlighted the inherent limitations and challenges associated with each model, ranging from overfitting and computational demands to issues of interpretability and generalization. These challenges are not merely obstacles but rather opportunities for further research and development in the field of machine learning and artificial intelligence. They prompt a continuous quest for more efficient, transparent, and robust

models that can navigate the complexities of real-world data and applications.

Looking forward, the field of image classification stands on the cusp of significant advancements. Emerging techniques in deep learning, such as attention mechanisms and transformer models, offer promising avenues for addressing the limitations of current models. Additionally, the integration of unsupervised and semi-supervised learning methods presents an opportunity to leverage vast amounts of unlabeled data, potentially overcoming the challenges of data scarcity and annotation.

In conclusion, the comparative analysis of CNNs, ResNets, and LSTMs in image classification has not only showcased the strengths and weaknesses of these models but also illuminated the path forward. As we continue to explore and innovate, the evolution of machine learning models will undoubtedly enhance their efficacy, interpretability, and application, leading to groundbreaking advancements in image classification and beyond. The journey of discovery and improvement in artificial intelligence is an ongoing process, driven by challenges, inspired by limitations, and propelled forward by the relentless pursuit of knowledge and understanding

References

[1] Othman, A., & El Ghoul, O. (2021). Intra-linguistic and extra-linguistic annotation tool for the "Jumla Dataset" in Qatari sign language. 2021 8th International Conference on ICT & Accessibility (ICTA). IEEE. https://doi.org/10.1109/ICTA54582.2021.9809778

[2] Papatsimouli, M., Kollias, K.-F., Lazaridis, L., Maraslidis, G., Michailidis, H., Sarigiannidis, P., & Fragulis, G. F. (2022). Real Time Sign Language Translation Systems: A review study. 2022 11th International Conference on Modern Circuits and Systems Technologies (MOCAST). IEEE. https://doi.org/10.1109/MOCAST54814.2022.9837666

[3] Boondamnoen, M., Thongsri, K., Sahabantoegnsin, T., & Woraratpanya, K. (2023). Exploring LSTM and CNN architectures for sign language translation. 2023 15th International Conference on Information Technology and Electrical Engineering (ICITEE). IEEE. https://doi.org/10.1109/ICITEE59582.2023.10317660

[4] Yin, K. (2020). Sign Language Translation with Transformers. arXiv preprint arXiv:2004.00588. https://arxiv.org/abs/2004.00588

[5] Pandian, K., Mohd Razman, M. A., Mohd Khairuddin, I., Abdullah, M. A., Ab Nasir, A. F., & Mat Isa, W. H. (2023). Sign Language Recognition using Deep Learning through LSTM and CNN. Mekatronika, 5(1), 67-71. https://doi.org/10.15282/mekatronika.v5i1.9410.

[6] Sharma, S., & Singh, S. (2021). Vision-based hand gesture recognition using deep learning for the interpretation of sign language. Expert Systems With Applications, 182, 115657. https://doi.org/10.1016/j.eswa.2021.115657.

[7] Abu-Jamie, T. N., & Abu-Naser, S. S. (2022). Classification of Sign-Language Using Deep Learning by ResNet. International Journal of Academic Information Systems Research (IJAISR), 6(8), 25-34.

Author Information

Kamel Mojtaba Kamel, Faculty of Computing and Informatics, Multimedia University
Amin Ahmed Alawad, Faculty of Computing and Informatics, Multimedia University
Obai Ali Abdelrahman, Faculty of Computing and Informatics, Multimedia University
Ahmed Abdulkhaleq Alnasri, Faculty of Computing and Informatics, Multimedia University