**Yarmouk University**



**College of Information Technology**

**and**

**Computer Science**

**Department of Information Systems**

**Data Science & Artificial Intelligence program**

**[From ASL TO ALL LANGUAGES]**

**Students' name:**

**Ahmed AL-Jarrah, Sewar Qudah, Sura Keelani**

**Supervisor:**

**Dr. Enas Alikhashashneh**

**FIRST TERM 2024**

**Table of content**

|  |  |  |
| --- | --- | --- |
|  | Abstract | 4 |
|  | Introduction | 4 |
|  | Project Objectives | 5 |
|  | Project overviwe | 5 |
|  | Literature Review | 6 |
|  | Methodology   * 1. Data Collection   2. Preprocessing   3. Techniques   4. Performance Metrics   5. Product | 13 |
|  | Results and Discussion | 16 |
|  | Limitations | 25 |
|  | Conclusion | 26 |
|  | Future Work | 27 |
|  | References | 27 |
|  | Appendix | 30 |

**List of Tables**

|  |  |  |
| --- | --- | --- |
| Table 1 | Literature reviews | 8 |
| Table 2 | Comparison table between models | 24 |

|  |  |  |
| --- | --- | --- |
| Figure 1 | Methodology flowchart | 13 |
| Figure 2 | Model accuracy & Loss (CNN) | 19 |
| Figure 3 | Model accuracy & Loss (VGG16) | 22 |
| Figure 4 | Model accuracy & Loss (RESNET50) | 21 |
| Figure 5 | Model accuracy & Loss (VGG19) | 20 |
| Figure 6 | Model accuracy & Loss (MOBILENET) | 23 |

**List of Figures**

**Abstract**

Effective communication is a vital aspect of human interaction, yet millions of deaf individuals face barriers due to limited accessibility to sign language interpretation. Fingerspelling, a core component of American Sign Language (ASL), plays a critical role in bridging linguistic gaps and fostering inclusivity. This project presents an AI-driven solution designed to decode ASL fingerspelling gestures into text and optional voice outputs, leveraging advanced computer vision techniques. By utilizing Convolutional Neural Networks (CNNs) and state-of-the-art pre-trained models such as MobileNet, the system achieves high accuracy and efficiency, making it suitable for real-world applications.

The project rigorously evaluates models like VGG16, VGG19, ResNet50, and MobileNet, with the latter emerging as the most effective due to its lightweight architecture and exceptional performance metrics, including 99.5% accuracy and a test loss of 0.01. Delivered through an intuitive web platform, this system empowers educators, learners, and the broader community by fostering accessibility and inclusivity. Looking ahead, this technology offers potential for expansion into additional sign languages, paving the way for a more interconnected and inclusive society where communication barriers are eliminated through the transformative power of AI.

1. **Introduction**

Communication is the cornerstone of human connection, enabling us to share ideas, emotions, and knowledge. For millions of deaf individuals worldwide, sign language is their primary means of interaction. Among the foundational aspects of sign language is fingerspelling, the manual representation of alphabetical letters, allowing users to spell words, names, and terms lacking standard signs. This skill not only bridges linguistic gaps but also serves as an essential entry point for sign language learners, offering them a practical and immediate way to express themselves.

However, a significant challenge remains: the gap between those fluent in sign language and those unfamiliar with it. This gap often leads to misunderstandings and exclusions in daily life, from classroom activities to professional and personal interactions. School teachers, for instance, see immense potential in incorporating fingerspelling into their lessons to enhance spelling classes and foster inclusivity. Yet, the lack of accessible tools to translate and interpret fingerspelling in real time limits the broader adoption and understanding of this invaluable communication tool.

To address this challenge, this project proposes an AI-powered system that leverages state-of-the-art computer vision techniques to decode American Sign Language (ASL) fingerspelling gestures. The system will translate these gestures into text, further enabling translation into other languages and optional voice outputs. At the heart of this solution are Convolutional Neural Networks (CNNs) and pre-trained models, celebrated for their exceptional capability to process and analyze visual data with precision. Delivered through a user-friendly web platform, this approach combines technical sophistication with accessibility, ensuring that users of all backgrounds can benefit from its capabilities.

This project aspires to bridge communication divide by transforming fingerspelling into a universally comprehensible medium. Empowering educators, learners, and the wider community, it promotes inclusivity and accessibility, making it easier for everyone to engage with sign language. In doing so, we aim to foster a world where communication barriers are dismantled, and every individual's voice, whether signed, spoken, or written, is heard and understood. Through the power of AI, we take a meaningful step toward creating a truly interconnected and inclusive society.

1. **Project Objectives**
   * Implement and apply theoretical knowledge and skills that we learned to develop an AI-based solution for communication challenges.
   * Create a user-friendly and intuitive platform that ensures smooth functionality and minimizes technical issues.
   * Address critical communication barriers for individuals who are unable to speak, offering a practical and impactful tool.
   * Design a scalable system that supports future enhancements, integrations, and broader use cases.
   * Reduce waiting times and eliminate miscommunication in service sectors, enhancing user and

provider satisfaction.

1. **Project Overview**

The AI-powered Website for Deaf Interaction project aims to address a significant real-world communication barrier faced by the deaf community. By leveraging artificial intelligence and cutting-edge computer vision techniques, this solution bridges the gap between signers and non-signers, promoting social inclusion and fostering a more accessible society.

The platform operates as a web-based application designed to recognize American Sign Language (ASL) gestures and translate them into text. The system further enhances communication by providing voice output options and multi-language text translation, enabling users to interact effectively in diverse settings. Using Convolutional Neural Networks (CNNs) and pre-trained models, the platform ensures high accuracy in gesture recognition and translation, even in real-time applications.

The design emphasizes usability, accessibility, and adaptability. With a focus on a seamless user experience, the interface is intuitive and easy to navigate, ensuring it accommodates users with varying levels of technical expertise. The solution is future-ready, with scalability for integrating advancements in AI and expanding its functionality beyond government sectors to education, healthcare, and personal use.

**4.Literature Review**

The paper presents a system that utilizes image processing, machine learning, and template-matching techniques to translate binary sign language into text and speech. Key algorithms include thresholding, color calibration, and coordinate mapping. The system effectively recognizes static gestures and converts them into textual and audio outputs but faces challenges such as lighting variations, difficulty recognizing dynamic gestures, and the need for recalibration under changing conditions. While no specific accuracy metrics are provided, the system relies on template matching for precise results. It uses a database created during training and aims to enhance communication between individuals with hearing impairments and the broader community through binary sign language, which is based on the number of open fingers in each gesture [1]

The paper on "Sign Language Recognition" focuses on developing systems to bridge communication barriers for the deaf community by interpreting sign language. It uses techniques such as data gloves, sensors, and vision-based methods like Principal Component Analysis (PCA) and mathematical models like Neural Networks (NN) and Hidden Markov Models (HMM) to improve accuracy. Studies have shown an accuracy of up to 90.8% in continuous sign recognition using techniques like DTW, with non-manual features incorporated for better performance. Key challenges include signer variability, occlusion, and data scarcity. Datasets like BSL, DGS, and DictaSign are used, focusing on American, British, and German Sign Languages [2]

The paper focuses on advanced techniques for sign language recognition using computer vision and statistical models. It employs multiple hypotheses tracking (MHT) to analyze hand movements, active appearance models (AAM) to extract facial expressions, and hidden Markov models (HMMs) for classification. While achieving high accuracy of up to 98% in controlled environments, challenges include handling individual variations, complex backgrounds, and occlusions. The study emphasizes languages such as German Sign Language (DGS) and British Sign Language (BSL), aiming to develop portable, user-independent systems [3]

The paper presents a system that uses Convolutional Neural Networks (CNN) and image processing techniques such as HSV color space transformation, dilation, erosion, and binary image segmentation for background removal and gesture recognition. The system achieves an accuracy of over 90% in recognizing 10 static gestures from American Sign Language (ASL) using TensorFlow and OpenCV libraries. A custom dataset of 2000 images, split into 80% for training and 20% for testing, was created. The system faces challenges such as sensitivity to lighting conditions and difficulty maintaining consistent hand positions during data collection. The study aims to enhance communication between hearing-impaired individuals and non-sign language users by building an efficient sign-language recognition model [4]

The paper explores the use of Convolutional Neural Networks (CNNs) for automatic sign language recognition, specifically recognizing 20 Italian gestures from the ChaLearn 2014 dataset. The system achieves a cross-validation accuracy of 91.7% and a test accuracy of 95.68%, demonstrating strong generalization to unseen users and environments. Preprocessing steps include noise reduction, cropping, and temporal segmentation, while the implementation leverages Python libraries such as Theano and PyLearn2. Challenges include handling noisy movements and optimizing gesture boundaries. The work contributes to addressing the communication barrier between the Deaf community and the hearing majority [5]

The paper explores sign language recognition using linguistic sub-units to analyze motion, location, and hand shape features. It employs Hidden Markov Models (HMMs) and Sequential Pattern Boosting (SP Boosting) to enhance recognition accuracy, achieving 76% in signer-independence tests and up to 99.9% in controlled environments. The study focuses on German Sign Language (DGS) and Greek Sign Language (GSL) using diverse datasets. Key challenges include the need for linguistically annotated data and handling variations in backgrounds and user-specific signing styles, to develop more generalized and user-independent systems [6]

The paper presents a system for recognizing static signs of Indian Sign Language (ISL) using Convolutional Neural Networks (CNN). The model was trained on a dataset of 35,000 images containing 100 static signs (letters, numbers, and common words) captured under diverse environmental conditions. The system achieved a high accuracy of 99.90% on grayscale images and 98.70% on colored images. Techniques such as Max-pooling, ReLU, and Softmax were utilized, alongside optimization algorithms like Adam and SGD. The research faced challenges in processing a wide variety of signs and the lack of ready-to-use datasets. This system is a step forward in facilitating communication with the hearing-impaired and holds the potential for future expansion to dynamic signs [7]

The paper provides a comprehensive review of hand gesture and sign language recognition techniques, utilizing algorithms such as Hidden Markov Models (HMMs) and Dynamic Time Warping (DTW) for dynamic gestures, as well as feature extraction methods like SIFT, SURF, PCA, and LDA. Classifiers like Artificial Neural Networks (ANN) and Support Vector Machines (SVM) were employed, achieving accuracy levels between 85% and 99%. Key challenges include variations in lighting, rapid movements, and the complexity of dynamic gestures. The study uses diverse datasets, including ASL and ArSL, along with sensor devices like Kinect and Leap Motion. The research aims to develop effective and reliable systems to improve interaction between the Deaf community and society through advanced technologies that integrate computer vision and sensors [8]

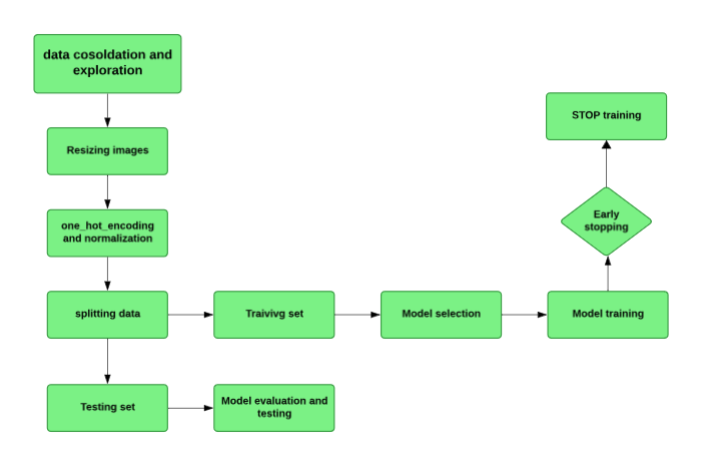
A system for sign language recognition was developed using machine learning techniques, specifically the **Random Forest** algorithm. Libraries such as **OpenCV** and **MediaPipe** were integrated for real-time hand gesture tracking and analysis. An interactive interface was built using **Tkinter**, while gestures were converted to speech using **Pyttsx3**. The model showed good performance for certain gestures but faced challenges with intricate and unfamiliar movements. The dataset included diverse hand gesture images that underwent preprocessing to enhance model performance. The study aims to bridge the communication gap for the deaf and mute by translating sign language into text or speech [9]

The paper introduces a system for recognizing American Sign Language (ASL) using a multi-headed Convolutional Neural Network (CNN) that integrates image processing and hand landmark detection. The proposed model achieved a test accuracy of **98.98%** by combining both inputs. Developed with TensorFlow, Keras, and OpenCV, the system utilizes the **ASL Finger Spelling** dataset, which includes over 65,000 RGB images of 24 static letters. Key challenges involve improving accuracy in noisy real-world environments and reducing dependency on hand landmarks. The research aims to provide an efficient, low-cost ASL recognition solution that does not require expensive sensors [10]

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Research number** | **Research name** | **Dataset used** | **Techniques** | **Results** |
| **[1]** | Intelligent Sign Language Recognition Using Image Processing | The system uses a custom database created during the training process, with no mention of external datasets. | |  | | --- | | Image processing, machine learning, template matching, text-to-speech conversion, thresholding. | |  | | |  | | --- | | Development of a system that translates binary sign language into text and speech, improving communication for individuals with disabilities. | |  | |
| **[2]** | Sign Language Recognition | |  | | --- | | BSL Corpus, DGS-Korpus, DictaSign, and custom datasets. | |  | | |  | | --- | | Data gloves, PCA, HMM, NN, DTW, PaHMMs, AAMs, hybrid approaches. | |  | | |  | | --- | | Improved accuracy by integrating manual and non-manual features and applying linguistic grammars. | |  | |
| **[3]** | Recent developments in visual sign language recognition | |  | | --- | | Video datasets: specific names not mentioned | |  | | |  | | --- | | MHT, AAM, HMMs, PDMs | |  | | |  | | --- | | Independent systems, single-camera setup, leveraging hand and facial features for better accuracy | |  | |
| **[4]** | Sign Language Recognition System Using  Convolutional Neural Network And  ComputerVision | |  | | --- | | The Custom dataset of 2000 images split into 80% for training and 20% for testing. | |  | | |  | | --- | | Convolutional Neural Networks (CNN), image processing with OpenCV, HSV color space transformation, dilation, erosion, binary segmentation. | | Convolutional Neural Networks (CNN), image processing with OpenCV, HSV color space transformation, dilation, erosion, binary segmentation. | | |  | | --- | | Successfully recognizes 10 static gestures from ASL with strong performance on static images. | | Successfully recognizes 10 static gestures from ASL with strong performance on static images. | |
| **[5]** | Sign Language Recognition using Convolutional Neural Networks | |  | | --- | | ChaLearn 2014 dataset (20 Italian gestures, 27 users, varied environments). | |  | | |  | | --- | | CNNs, sliding window temporal segmentation, preprocessing (noise reduction, cropping). | |  | | |  | | --- | | Generalizes well to unseen users and environments; ranked 5th in ChaLearn 2014 competition. | |  | |
| **[6]** | Sign Language Recognition using Sub-Units | |  | | --- | | 40-sign German Sign Language (DGS), 20-sign Greek Sign Language (GSL). | |  | | |  | | --- | | Linguistic sub-units, HMMs, SP Boosting. | |  | | |  | | --- | | User-independent systems, integration of motion, location, and hand shape features. | |  | |
| **[7]** | Deep learning-based sign language recognition system for static signs | |  | | --- | | 35,000 images including letters (23), numbers (0–10), and 67 common words. | | 35,000 images including letters (23), numbers (0–10), and 67 common words. | | |  | | --- | | CNN, Max-pooling, Softmax, ReLU, optimization algorithms (Adam, SGD, RMSProp). | |  | | |  | | --- | | Successfully recognized 100 static ISL signs with high accuracy using a diverse dataset. | |  | |
| **[8]** | A review of hand gestures and sign language recognition techniques | |  | | --- | | ASL, ArSL, and sensor-based data from Kinect and Leap Motion. | |  | | |  | | --- | | HMMs, DTW, SIFT, SURF, PCA, LDA. | |  | | |  | | --- | | High accuracy in static and dynamic gestures; improved performance via sensor integration. | |  | |
| **[9]** | SIGN LANGUAGE RECOGNITION | |  | | --- | | Diverse hand gesture data, preprocessed (e.g., resizing, grayscale conversion). | |  | | |  | | --- | | Random Forest, OpenCV, MediaPipe, Pyttsx3, Tkinter, Matplotlib, CNNs | |  | | |  | | --- | | Accurate real-time gesture recognition with a user-friendly interface. | |  | |
| **[10]** | Sign language recognition using  the fusion of image and hand  landmarks through multi‑headed  convolutional neural network | |  | | --- | | **ASL Finger Spelling** dataset with 65,000+ RGB images (24 static letters; j and z excluded due to motion requirements). | |  | | |  | | --- | | Multi-headed CNN combining image processing and hand landmarks with Dropout, Batch Normalization, and dynamic learning rate adjustment. | | Multi-headed CNN combining image processing and hand landmarks with Dropout, Batch Normalization, and dynamic learning rate adjustment. | | |  | | --- | | High accuracy in recognizing ASL static letters in complex backgrounds, showcasing efficiency and robustness. | |

**Table 1: literature reviews**

1. **Methodology:**

****

**Figure 1: Methodology flowchart**

**4.1: Dataset and exploration**

The dataset obtained from Kaggle consists of 87,000 images, making it a comprehensive resource for projects related to American Sign Language (ASL). It includes **29 classes**, covering all 26 letters of the English alphabet and 3 additional words: **SPACE**, **DELETE**, and **NOTHING**. Each class is represented by **3,000 images**, ensuring a balanced representation across all categories. The images in the dataset exhibit diversity in angles, lighting conditions, and backgrounds, providing a robust foundation for training machine learning models. This dataset is ideal for building systems capable of recognizing ASL signs with high accuracy

**4.2. Preprocessing:**

### 4.2.1 ****Resizing Images****

### The resize\_image function is responsible for resizing the images to a uniform size of 64x64 pixels. This ensures that all images in the dataset have the same dimensions, making them suitable for input into machine learning models. The function uses the resize method from a specific image processing library to scale the images without altering their aspect ratio, which is crucial for consistent model training.

**4.2.2 One-Hot Encoding of Labels:**

One-hot encoding is applied to the labels for the training, validation, and testing datasets using the to\_categorical function from Keras. One-hot encoding is a method of converting categorical labels into a binary matrix representation, where each class label is represented as a vector of zeros with one in the position corresponding to the class. This is necessary because most machine learning models (especially neural networks) require the target labels to be in a one-hot encoded format.

**5. Splitting data**:

To effectively prepare the dataset for training and evaluation, the data is split into three subsets: training, validation, and testing. The full dataset, consisting of 87,000 samples, is initially divided into 70% for training (60,900 samples) and 30% for testing and validation (26,100 samples). The testing and validation portion is further split equally, allocating 15% of the total dataset to each (13,050 samples for validation and 13,050 samples for testing). This split ensures that the model has a large training set to learn from, a separate validation set for tuning and hyperparameter optimization, and an independent testing set for evaluating final performance, enabling robust and unbiased model evaluation.

**5. Model selection:**

**5.1 - Customized CNN**: A customized Convolutional Neural Network (CNN) is tailored specifically for the task at hand, allowing us to optimize and control the architecture for detecting ASL signs. By designing a model from scratch, we can focus on the nuances of ASL signs, addressing challenges like similar letter signs. This flexibility ensures the network adapts to our dataset and the requirements of our project**.**

**5.2. VGG16:** VGG16 is a popular pre-trained deep learning model known for its simplicity and strong performance in image classification tasks. Its deep architecture, with small receptive fields and uniform convolutional layers, is ideal for recognizing the underlying patterns in ASL signs. Using VGG16 helps leverage its robust feature extraction capabilities, particularly for detecting subtle variations in hand shapes.

**5.3 VGG19:** Like VGG16, VGG19 extends the architecture with additional layers, offering a deeper network. This depth enables the model to capture finer details and more complex patterns, making it suitable for distinguishing between visually similar ASL letters. Its proven track record in image-related tasks adds confidence to its application in our project.

**5.4 ResNet50**: ResNet50, a deep residual network, addresses the problem of vanishing gradients in deep networks using residual connections. Its ability to train very deep architecture effectively makes it well-suited for AS detection, where subtle variations in gestures can benefit from deeper feature extraction. Its robustness and efficiency make it a valuable choice for comparison in our project.

**5.5 MobileNet**: This is a lightweight deep-learning model optimized for mobile and embedded systems, making it an excellent choice for real-time ASL detection applications (as we tend to use later). Its focus on efficiency and reduced computational requirements aligns with the project's resource constraints while maintaining high accuracy. MobileNet’s adaptability ensures it can handle the nuances of AL signs without requiring extensive computational resources.

**6. Early Stopping** is a regularization technique that prevents overfitting and saves computational resources by halting model training when performance on the validation set no longer improves. It monitors a specified metric, such as validation loss (monitor='val\_loss'), which measures the model's error on the validation dataset. Training stops if the metric fails to improve for a defined number of epochs (patience=3), allowing the model to recover from minor fluctuations without premature termination. With verbose=1, it provides feedback during training by notifying when Early Stopping is triggered, enhancing visibility into the process.

1. **Results and Discussion**

**The evaluation metrics that we used:**

1. **Accuracy (Training and Testing Datasets):**

Accuracy measures the proportion of correctly predicted ASL signs out of the total predictions. On the training dataset, accuracy reflects how well the model has learned the patterns in the data, while on the testing dataset, it shows the model's ability to generalize to unseen data. In the context of ASL detection, high accuracy indicates that the model can reliably identify most signs, which is crucial for effective communication. This metric is a fundamental choice because it provides a direct understanding of the model's overall performance, ensuring it meets the practical demands of real-world ASL recognition.

**2. Validation Loss (Training Phase):**

Validation loss evaluates how well the model predicts on unseen validation data during training. It measures the discrepancy between predicted outputs and actual labels. Monitoring validation loss helps us detect overfitting, ensuring the model does not merely memorize training data but instead learns to generalize. For ASL detection, a decreasing validation loss indicates that the model is becoming proficient at recognizing signs while maintaining robustness against new data, a critical requirement for ensuring consistent performance in diverse communication scenarios.

**3. Macro Average:**

Macro averaging computes the unweighted mean performance across all classes, treating each ASL sign equally regardless of its frequency in the dataset. This metric is particularly valuable in ASL detection, where signs with similar appearances or lower frequencies may be harder to identify. By emphasizing equal importance for all signs, macro averaging ensures that the model performs well not only on common signs but also on less frequent or visually similar ones, supporting inclusivity and reliability in communication.

**4. Weighted Average:**

Weighted averaging accounts for class imbalance by weighing each class's performance according to its frequency in the dataset. In the context of ASL detection, where some signs might appear more often than others, this metric provides a realistic assessment of the model's performance. It ensures that the evaluation reflects practical usage scenarios, emphasizing the accurate recognition of both common and rare signs proportionally.

**5. Confusion Matrix:**

The confusion matrix offers a detailed breakdown of the model's predictions, showing true positives, false positives, true negatives, and false negatives for each class. For ASL detection, it helps identify specific signs that the model struggles to distinguish, such as letters with visual similarities. This granular insight enables targeted improvements in the model, ensuring that even the most challenging signs are recognized with high accuracy. The confusion matrix is indispensable for diagnosing misclassifications and refining the model's ability to handle real-world complexities.

**6. Precision:**

Precision measures the proportion of correctly predicted instances for a class out of all predictions made for that class:

High Precision: Most predicted instances of a class are correct.

**7. Recall:**

Recalling measures the proportion of correctly predicted instances for a class out of all actual instances of that class:

High Recall: The model identifies the truest instances of the class.

**8. F1-Score:**

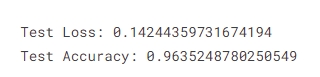
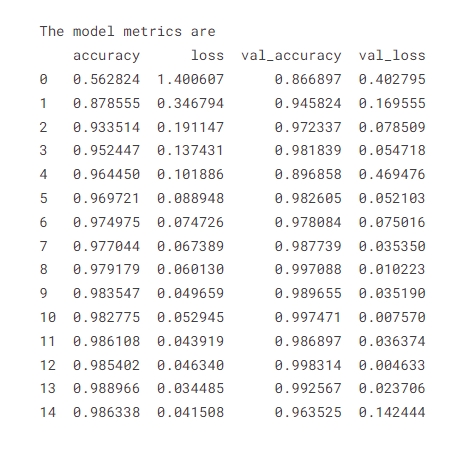
The F1-score is the harmonic means of precision and recall, providing a balanced evaluation:

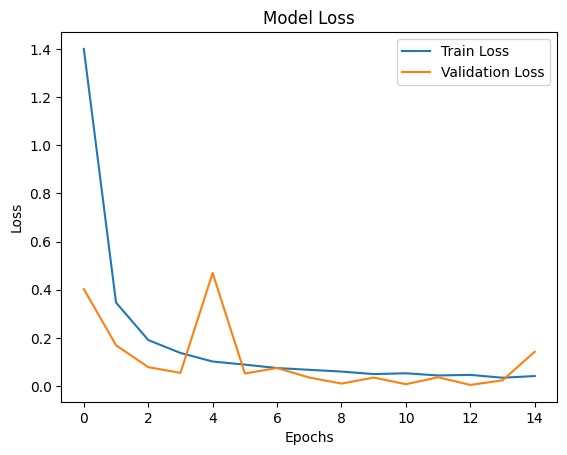
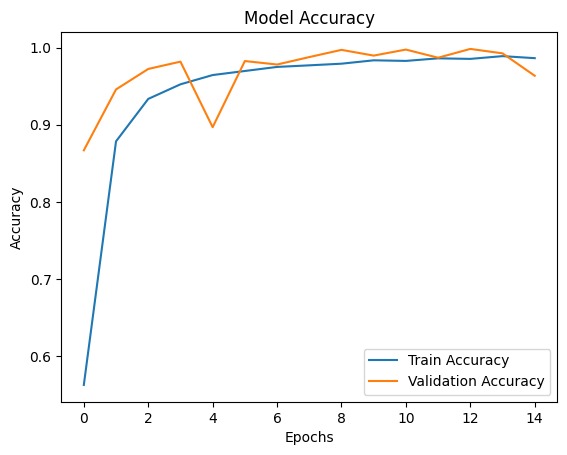
High F1-Score: Indicates strong overall performance for the class, balancing false positives and false negatives.

**9. Support:**

Support indicates the number of true instances for each class in the dataset.

**CNN**

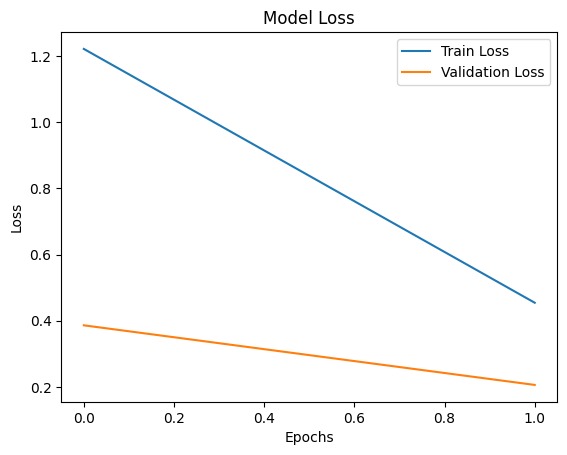
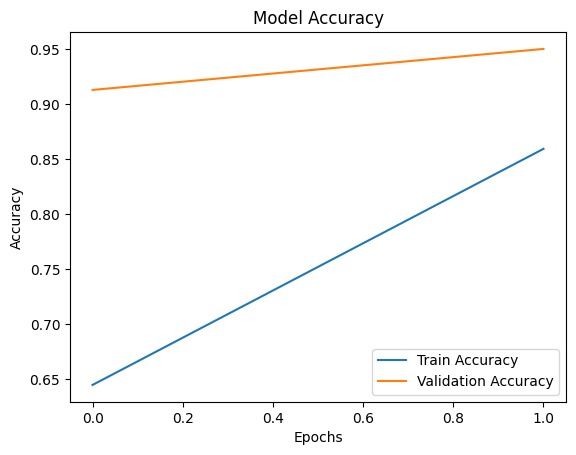
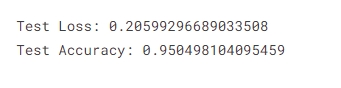
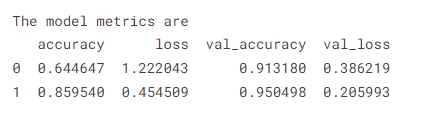




**Figure 2: Model accuracy & Loss (CNN)**

The results for the Convolutional Neural Network (CNN) model indicate robust performance. Over 15 epochs, the model achieves an impressive test accuracy of approximately 96.35%, with a corresponding low test loss of 0.142. The training and validation accuracy graphs demonstrate rapid improvement initially, stabilizing as the epoch progress. Similarly, the loss graphs depict a sharp decline early on, eventually plateauing. The final performance metrics suggest that the model generalizes well, maintaining high accuracy and low loss across both training and validation datasets. This consistency highlights the model's efficacy in learning and predicting accurately.

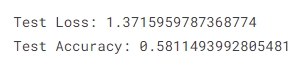
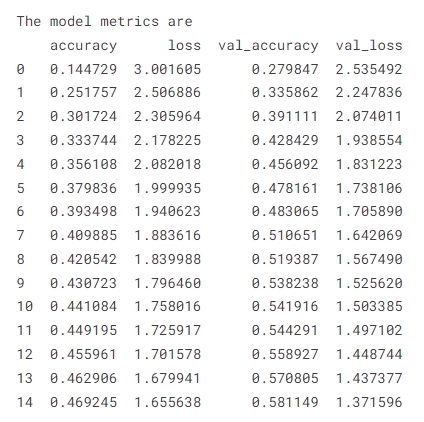
**VGG19:**

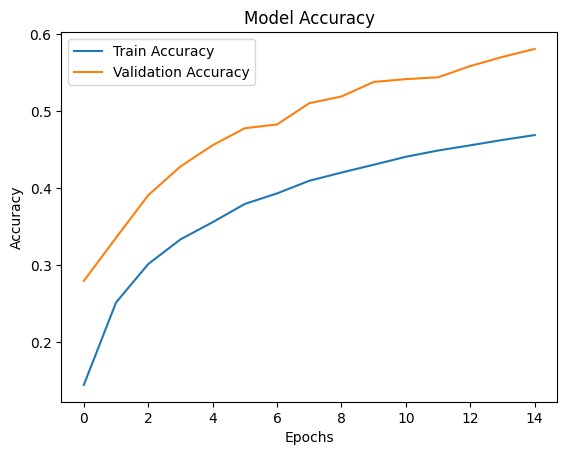
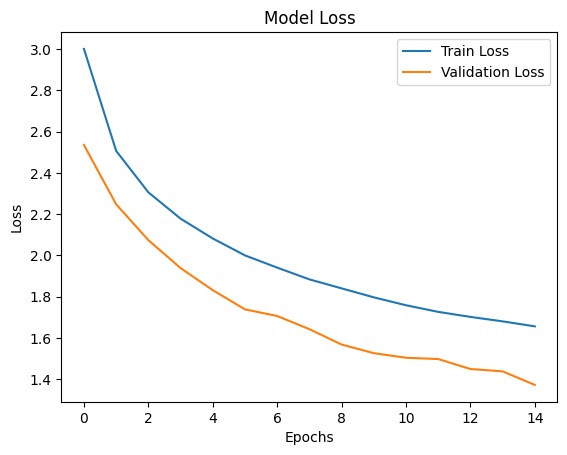


**Figure 3: Model accuracy & Loss (VGG19)**

The VGG19 model results indicate excellent performance. The test accuracy is approximately 95.05%, and the test loss is 0.2059. Over the epoch, the model accuracy graph shows a steady increase in training accuracy, with the validation accuracy also showing a significant increase, albeit slightly lower than the training accuracy. The loss graph shows a sharp decrease in both training and validation losses, stabilizing towards the end. This suggests that the model effectively learns the patterns in the data, achieving high accuracy and low loss, and generalizes well across both training and validation datasets.

**RESNET50**

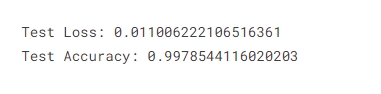
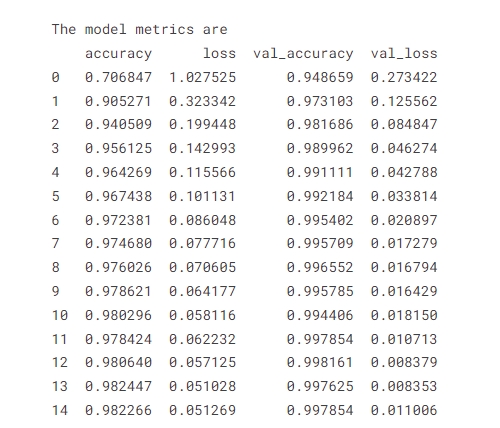
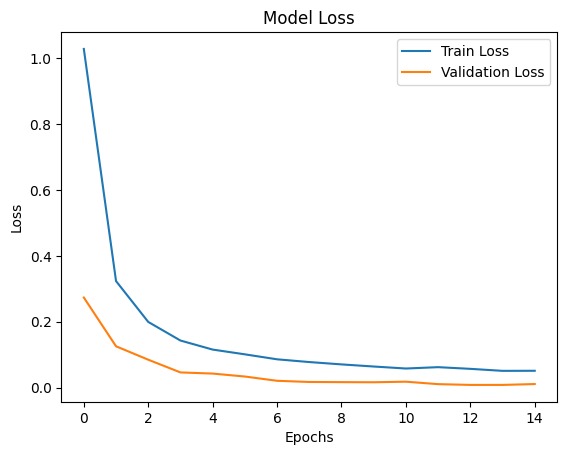
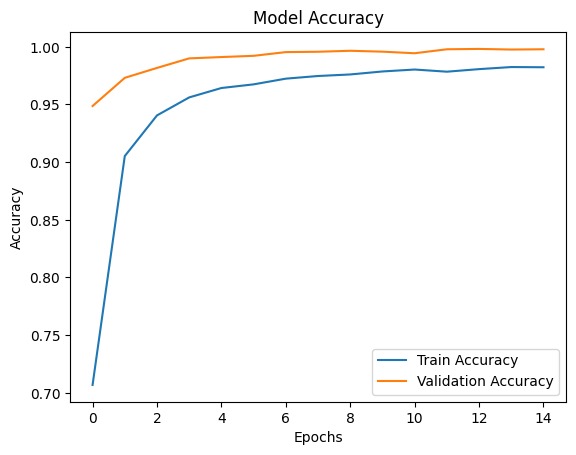


 **Figure 4: Model accuracy & Loss (RESNET50)**

The RESNET50 model shows suboptimal performance with a final test loss of **1.3716** and test accuracy of **0.5811**. Training accuracy remains low at **0.4692** by the 14th epoch, and validation accuracy improves gradually but stays around **0.5811**. Both training and validation loss decrease over epochs but do not reach optimal levels, indicating potential underfitting.

The graphs confirm this, showing decreasing loss without stabilization and gradually improving but low accuracy. To enhance performance, consider hyperparameter tuning, increasing model complexity, or improving the quality and quantity of training data.

**VGG16**



**Figure 5: Model accuracy & Loss (VGG16)**

The results for the VGG16 model indicate very strong performance. Over 15 epochs, the model achieves an impressive test accuracy of approximately 99.79% and a low test loss of 0.0111. The training and validation accuracy graphs show a consistent and significant improvement, with both metrics near 100%. Similarly, the loss graphs show a steady decline for both training and validation datasets, reaching very low values by the end.

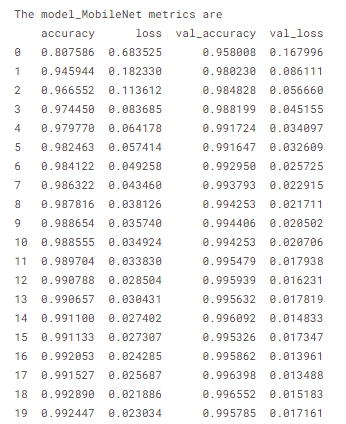
The table of metrics highlights the following key points:

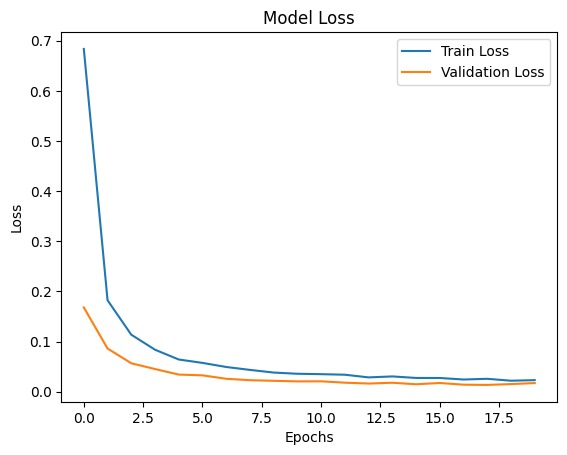
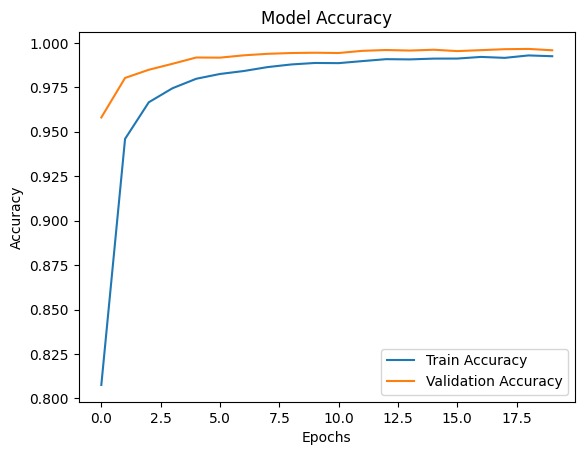
- Epoch 0: Initial training accuracy starts at 76.68%, with a validation accuracy of 94.87%.

- Epoch 14: Final training accuracy reaches 98.23%, with validation accuracy closely matching at 99.79%.

Overall, these results suggest that the VGG16 model is highly effective, with excellent generalization capabilities and a very high accuracy rate across both training and validation datasets. This consistency and low loss value indicate the model's efficacy in learning and predicting accurately.

**MOBILENET**

****

**Figure 6: Model accuracy & Loss (MOBILENET)**

The results for the MobileNet model suggest a very strong performance. Over 20 epochs, the model achieves an impressive test accuracy of approximately 99.58% and a low test loss of 0.0172. The training and validation accuracy graphs demonstrate rapid and consistent improvement, with both nearing 100% by the final epochs. Similarly, the loss graphs show a steady decline for both training and validation datasets, reaching very low values by the end.

Key points from the table of metrics include:

• Epoch 0: Initial training accuracy starts at 89.76%, with a validation accuracy of 95.08%.

• Epoch 19: Final training accuracy reaches 99.24%, with validation accuracy closely matching at 99.58%.

Overall, these results indicate that the MobileNet model is highly effective, with excellent generalization capabilities and a very high accuracy rate across both training and validation datasets. This consistency and low loss value suggest the model's efficacy in learning and predicting accurately.

|  |  |  |
| --- | --- | --- |
| **Model Name** | **Training accuracy** | **Testing accuracy** |
| **CNN** | 98.63% | 96.35% |
| **VGG19** | 85.95% | 95.05% |
| **RESNET50** | 46.92% | 58.11% |
| **VGG16** | 98.23% | 99.79% |
| **MOBILENET** | 99.24% | 99.58% |

**Table 2: Comparison table between models**

1. **Limitations**

* Previous Project Transition: Transitioning from a previously abandoned project has consumed valuable time.
* Sign Similarities: Certain letter signs have visual similarities, making it challenging for the model to differentiate them accurately.
* Limited Resources: The lack of sufficient computational resources has restricted the ability to train and optimize models effectively, especially given the project's high demand for processing power.
* High Data Volume: Handling and processing a large dataset (96,000 images) have posed challenges, leading to slower progress and additional technical complexities.
* Platform Limitations: Reliance on free or limited computational platforms has resulted in session crashes and interruptions, significantly hindering the development and testing processes

**9. Conclusion**

This project represents a significant step forward in leveraging deep learning to address the communication challenges faced by the deaf community. By exploring and comparing multiple advanced models, including Customized CNN, VGG16, VGG19, ResNet50, and MobileNet, we have rigorously evaluated their performance to identify the most suitable solution for ASL detection. Among these models, MobileNet has emerged as the optimal choice, demonstrating compatibility with resource-constrained environments, high accuracy, and reliable predictive capabilities. Its lightweight architecture ensures efficient processing without compromising performance, making it ideal for real-world deployment.

The evaluation metrics have shown MobileNet’s superior results, with accuracy reaching 99.5% and Test loss at 0.01. These outcomes confirm their reliability in detecting ASL signs across diverse scenarios. Moving forward, this system holds the potential for expansion to include additional sign languages and integration into user-friendly applications, furthering its role in bridging the gap between the deaf community and the hearing world. This project not only demonstrates the power of AI in solving real-world challenges but also reinforces our commitment to creating inclusive, impactful solutions.

**10. Future Work**

1. Multilingual Support:

Expand the platform to support multiple sign languages, increasing

accessibility and inclusiveness.

2. Advanced AI Integration:

Incorporate state-of-the-art AI models for better performance and accuracy.

3. Sector Expansion:

Extend the application to other sectors, such as healthcare and education, for

wider impact.

4. Mobile Application Development:

Develop a mobile application to enhance accessibility, allowing users to interact

with the system anytime and anywhere.

1. **References**

**[1]** Sawant Pramada, Deshpande Saylee , Nale Pranita, Nerkar Samiksha , Mrs.Archana S. Vaidya

2013

Intelligent Sign Language Recognition Using Image Processing

<https://d1wqtxts1xzle7.cloudfront.net/30970559/H03224551-libre.pdf?1392205841=&response-content-disposition=inline%3B+filename%3DIOSR_Journal_of_Engineering_IOSR_JEN.pdf&Expires=1735147092&Signature=EgLvqlFsk7uDErHrSq9b2QdEOLZNzvcq7DPB0iA~DFk5VAavVDSzMkwDPpNKpq~P-mIoiew7usEfNikUwvJ4f6g1dOmi0-uV6gpwziR8LbcAImGXfNpcwAVkzDGOhiHMpgNAk7rVyTFJELzkS2UbE65uA6zcqX8yvv1OHaQC5YvxVd0fArV7o8A5rcHiKAOG9V20sfw0d6V0Xtu3bbzggGTvowOI7rkOjhE1lS9PJAMVN-pxO3ugqivhoKjP7ttvcAjRXTzwiOyhSH8Nq1G8lVOwFr3IyHOU3dspi9VbMJL9KimJQyxQGIUoDV3U5a1KTvfvrqh5zQNNDULVCh5GUw__&Key-Pair-Id=APKAJLOHF5GGSLRBV4ZA>

45-51

[2] Helen Cooper, Brian Holt and Richard Bowden

2011

Sign Language Recognition

<https://link.springer.com/chapter/10.1007/978-0-85729-997-0_27>

539–562

[3] Ulrich von Agris Æ Jo¨rg Zieren Æ Ulrich Canzler Æ Britta Bauer Æ Karl-Friedrich Kraiss

2007

Recent developments in visual sign language recognition

<https://link.springer.com/article/10.1007/s10209-007-0104-x>

323-362

[4] Romala Sri Lakshmi Murali, L.D.Ramayya , V. Anil Santosh

2022

Sign Language Recognition System Using Convolutional Neural Network And ComputerVision

<https://ijeiat.com/images/sliders/92dc4915b9487f589cfe29a2d410cc0a.pdf>

2582-1431

[5] Lionel Pigou, Sander Dieleman, Pieter-Jan Kindermans, Benjamin Schrauwen Ghent University, ELIS, Belgium

2015

Sign Language Recognition using Convolutional Neural Networks

<https://link.springer.com/chapter/10.1007/978-3-319-16178-5_40>

572–578

62

[6] Helen Cooper , Eng-Jon Ong , Nicolas Pugeault , Richard Bowden

2012

Sign Language Recognition using Sub-Units

<https://www.jmlr.org/papers/volume13/cooper12a/cooper12a.pdf>

2205-2231

[7] Ankita Wadhawan 1 • Parteek Kumar1

2020

Deep learning-based sign language recognition system for static signs

https://link.springer.com/article/10.1007/s00521-019-04691-y

7957–7968

[8] Ming Jin Cheok1 , Zaid Omar1 , Mohamed Hisham Jaward2

2019

A review of hand gesture and sign language recognition techniques

<https://link.springer.com/article/10.1007/s13042-017-0705-5>

131–153

[9] Mr. G. Ravi Kumar, K. Swathi, P. Sujana Priya, Ch. Sai Prasanna Kumar Reddy, A. Sai Sreeram, B. Dhana Lakshmi

2024

SIGN LANGUAGE RECOGNITION

<https://ieeexplore.ieee.org/abstract/document/7507939>

[10] Refat Khan Pathan , Munmun Biswas, Suraiya Yasmin , Mayeen Uddin Khandaker,Mohammad Salman , Ahmed A. F. Youssef

2023

Sign language recognition using the fusion of image and hand landmarks through the multi‑headed convolutional neural network.

<https://www.nature.com/articles/s41598-023-43852-x.pdf>

1. **Appendix:**

Dataset link:

https://www.kaggle.com/datasets/grassknoted/asl-alphabet

Code link:

https://www.kaggle.com/code/ahmedmahmoudaljarrah/da499-ahmed-m-jarrah