IJPAS ISSN: 2149-0910 | INTERNATIONAL JOURNAL PURE AND APPLIED SCIENCES

Vol. 11(2); 393 - 409 (2025)
DOI: 10.29132/ijpas.1637971

Research Article

Sign Language Recognition with Ensemble Learning and Bayesian Optimization: A Deep Learning-Based Approach

Andaç Fındıkçı[1], Musa Balcı[2], Hüseyin Aydilek[3,*] and Mustafa Yasin Erten[4]

Received: 11.02.2025

Accepted: 03.10.2025

[1]Kırıkkale University, Faculty of Engineering and Natural Sciences, Departmant of Electrical and Electronics Engineering, Kırıkkale, Türkiye, andac06148@gmail.com

[2]Kırıkkale University, Faculty of Engineering and Natural Sciences, Departmant of Electrical and Electronics Engineering, Kırıkkale, Türkiye, musabalci833@gmail.com

[3]Kırıkkale University, Faculty of Engineering and Natural Sciences, Departmant of Electrical and Electronics Engineering, Kırıkkale, Türkiye, huseyinaydilek@kku.edu.tr

[4]Kırıkkale University, Faculty of Engineering and Natural Sciences, Departmant of Electrical and Electronics Engineering, Kırıkkale, Türkiye, mustafaerten@kku.edu.tr

*Corresponding author

Abstract: Communication has undergone a continuous evolution as one of the most fundamental needs in human history. Initially, communication was established through body language and gestures, but it became more complex over time with the development of spoken language. The invention of writing marked a revolutionary milestone in the history of communication. However, this rapid advancement also brought about communication challenges. In today's world, numerous studies focus on addressing these issues and finding effective solutions. Technological advancements and artificial intelligence hold significant potential for solving communication problems. Notably, the difficulties in communicating with individuals who are hearing impaired have become a prominent area of focus. In this study, a model was developed using artificial intelligence algorithms to facilitate communication through sign language, specifically detecting American Sign Language. The model was created using deep learning architectures such as InceptionV3, DenseNet169, and VGG16, and trained on a dataset sourced from Kaggle. The results were combined using the ensemble learning method. The performance of the models was optimized through Bayesian search optimization algorithm and evaluated using metrics derived from confusion matrices. The findings revealed that ensemble learning models demonstrated superior performance, indicating that this model could serve as an effective tool in improving communication with hearing-impaired individuals.

Keywords: ensemble learning; bayesian hyperparameter optimization; InceptionV3; Dense-Net169; VGG16

Topluluk Öğrenmesi ve Bayes Optimizasyonu ile İşaret Dili Tanıma: Derin Öğrenme Temelli Bir Yaklaşım

Özet: İletişim, insanlık tarihinin en temel ihtiyaçlarından biri olarak sürekli bir evrim geçirmiştir. İlk dönemlerde beden dili ve jestlerle kurulan iletişim, zamanla konuşma dilinin gelişmesiyle daha karmaşık bir yapıya bürünmüştür. Yazının icadı ise iletişimde

devrim niteliğinde bir adım olmuştur. Ancak, bu hızlı gelişim iletişim sorunlarını da beraberinde getirmiştir. Günümüz dünyasında, bu problemler üzerine birçok çalışma yapılmakta ve çözüm yolları aranmaktadır. Teknolojik gelişmeler ve yapay zekâ, iletişim sorunlarına çözüm üretme potansiyeli taşımaktadır. Özellikle işitme engelli bireylerle iletişimde yaşanan zorluklar bu alanda öne çıkmaktadır. Bu çalışmada, işaret dili ile iletişimi kolaylaştırmak amacıyla yapay zekâ algoritmaları kullanılarak Amerikan İşaret Dili'ni tespit eden bir model geliştirilmiştir. InceptionV3, DenseNet169 ve VGG16 gibi derin öğrenme mimarileri kullanılarak oluşturulan model, Kaggle veri seti üzerinde eğitilmiş ve sonuçlar ensemble learning yöntemiyle birleştirilmiştir. Model performansları, Bayes optimizasyonu ile optimize edilmiş ve karmaşıklık matrislerine dayanan metriklerle değerlendirilmiştir. Sonuçlar, ensemble learning modellerinin daha yüksek performans gösterdiğini ortaya koymuş, bu modelin işitme engelli bireylerle iletişimde kullanılabilecek etkili bir araç olabileceği sonucuna ulaşılmıştır.

Anahtar Kelimeler: topluluk öğrenmesi; bayes hiperparametre optimizasyonu; InceptionV3; DenseNet169; VGG16

1. Introduction

In today's world, communication is not only a fundamental right of every individual but also a crucial element for the healthy functioning of society. Sign language, developed to enable effective communication between hearing-impaired individuals and society, presents certain challenges in terms of learning and application. In this language, where each letter and word is represented by hand and finger movements corresponding to each alphabet letter, the clarity and accuracy of gestures are essential [1]. Incorrect usage of gestures can lead to confusion or misunderstandings. Due to advancements in artificial intelligence, people can now communicate easily through sophisticated translation applications, even without knowing the other person’s language. With the development of deep learning and machine learning algorithms, Natural Language Processing (NLP) and image processing algorithms now provide highly accurate translations at near-human speech speed [2]. Typically, these translations facilitate communication between the languages of different countries. However, the absence of a translation application for communication between hearing-impaired and non-hearing-impaired individuals has motivated this research.

Various studies in literature have aimed to address this issue. In this study, recent studies from the last five years were examined. Abu-Jamie and Abu-Naser [3] used the MobileNet deep learning model to recognize sign language across a 29-class dataset. In a subsequent study, the authors [4] developed a deep learning model that translates sign language into text or speech. Their study, based on the Visual Geometry Group (VGG16) model and 43,500 images, produced more successful results than their previous studies. Murali et al. [5] developed a sign language recognition system using American Sign Language (ASL), achieving over 90% accuracy through the application of computer vision techniques and deep learning algorithms. Pigou et al. [6] proposed a model utilizing both a deep learning architecture and Microsoft Kinect to recognize 20 Italian sign language gestures. Using Convolutional Neural Network (CNN) architecture for feature extraction, the model reached a cross-validation accuracy of 91.7%. Daroya et al. [7] proposed a DenseNet-based model for recognizing ASL letters. This model can recognize real-time hand movements from webcam input and classify static hand poses in RGB images with 90.3% accuracy. Nareshkumar and Jaison [8] developed a lightweight neural network architecture for

mobile classification of ASL and Indian Sign Language letters. Despite operating in real-time and on low-resource devices, this MobileNet-based application achieved an accuracy rate of 98.77%. Bhattacharya et al. [9] utilized machine learning algorithms such as Support Vector Machine (SVM), K-Nearest Neighbors (KNN), Logistic Regression, and Naive Bayes to recognize finger alphabet gestures in images. A comparative analysis demonstrated that SVM outperformed the other machine learning methods. Hasan et al. [10] achieved an accuracy of 97.62% in ASL character recognition using the MNIST dataset and a CNN architecture. In their study, Barbhuiya et al. [11] modified pre-trained CNN models, such as AlexNet and VGG16, to recognize ASL letters and numbers and used multi-class SVM for classification. Amrutha and Prabu [12] developed a hand-motion-based recognition model that combines feature extraction using a convex hull with classification via the KNN algorithm to create an automatic sign language recognition system. While the model achieved 65% accuracy in a controlled environment with four participants, another approach in the same study attained 99.82% classification accuracy using a low-cost CPU. Abdul et al. [13] proposed a real-time deep learning model for Arabic sign language classification. This model aims to extract both spatial and temporal features, using an attention-based Inception model for spatial features and a Bidirectional Long Short-Term Memory (BiLSTM) model for temporal features. The model demonstrated high accuracy across various datasets, excelling in both static and dynamic signs, while achieving efficient performance with fewer layers and parameters. Leth [14] in his thesis, Leth combined ensemble learning and virtual reality technology to develop a Danish Sign Language recognition system, using vector encoding for gesture recognition and an n-gram linear classifier for NLP. Despite achieving only 41.5% accuracy, the UI's usability indicated room for improvement. Similarly, Kothadiya et al. [15] proposed an Explainable Artificial Intelligence (XAI)-supported model for sign language recognition, incorporating a ResNet50 and self-attention module-based ensemble learning model, which achieved 98.20% accuracy. Moreover, SHAP-based methods were used to make the model's predictions interpretable. Öztürk et al. [16] proposed a deep learning model based on VGG16 and ResNet-50 for COVID-19 diagnosis; this model achieved an accuracy of 88.06% using a Multi-Layer Perceptron (MLP) structure optimized with the whale optimization algorithm . Benbakreti et al. [17] utilized pre-trained models such as GoogleNet and ResNet-18 to classify eye diseases from fundus images; GoogleNet achieved the best performance with a 92.7% accuracy rate using the Stochastic Gradient Descent with Momentum (SGDM) optimizer.

Despite efforts to address this issue with deep learning methods, no studies in literature have applied both hyperparameter optimization and ensemble learning to enhance the performance of these methods. In this study, the proposed image classification model was created using InceptionV3, DenseNet169, and VGG16 algorithms and was trained individually on a dataset containing images of American Sign Language letters. An ensemble learning model was then created by combining the separately trained image classification algorithms, and Bayesian optimization was used for hyperparameter optimization to assess the impact of hyperparameters on training. To identify the best performance, two different ensemble learning models were trained—one with and one without Bayesian optimization. The experimental outcomes aim to investigate the effect of the ensemble learning model on image classification, as well as the impact of Bayesian optimization and fine-tuning on model performance.

2. Materials and Methods

2.1. Materials

In this study, ASL classification was conducted using the InceptionV3, DenseNet169, and VGG16 algorithms on a dataset obtained from Kaggle [18]. The dataset comprises 24 classes, each representing one of the 24 letters of the ASL alphabet, with 1,000 images per class corresponding to a specific letter. Each image in the dataset contains static hand gestures representing a particular letter. However, the letters 'J' and 'Z', which are represented by dynamic hand movements in the ASL fingerspelling alphabet, are excluded due to their dynamic nature, which is not compatible with static image classification. Consequently, the dataset focuses solely on static hand shapes.

To ensure high variability, the images include a range of hand sizes, positions, and orientations, enhancing the model's ability to generalize to real-world scenarios. Furthermore, the dataset exhibits a balanced class distribution, with an equal number of 1,000 images per class. This characteristic prevents any single class from being overrepresented during the training process, thereby enabling the model to achieve balanced performance across all classes.

2.2. Methods

In this study, the sign language recognition performance of deep learning models were analyzed using a dataset representing 24 letters of the English alphabet in sign language. The deep learning architectures employed in the study include InceptionV3, DenseNet169, and VGG16. The dataset was preprocessed using resized to 224x224 pixels and normalization techniques, including pixel value scaling to the [0,1] range and image resampling to ensure uniform input dimensions, followed by Bayesian optimization to determine the optimal hyperparameters for each model.

2.2.1. VGG16

In the VGG16 architecture, the network progresses from convolutional and pooling layers to fully connected layers. Each convolutional layer reduces the complexity of the image while enabling the discovery of deeper features. Max-pooling layers reduce dimensions while preserving essential features, thus enhancing the overall performance of the network. This architecture allows VGG16 to perform complex tasks effectively. Figure 1 illustrates the VGG16 architecture [19].

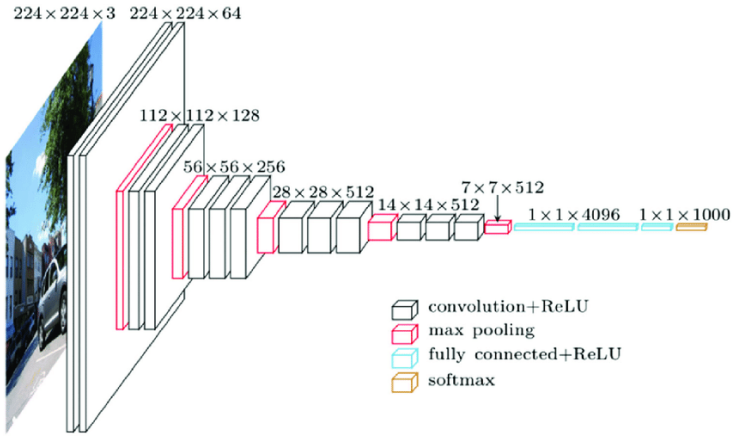

Figure 1. Structure of VGG16 [20]

2.2.2. InceptionV3

The InceptionV3 model is a 48-layer pre-trained convolutional neural network capable of processing images of various sizes and resolutions, making it suitable for large datasets. It utilizes a specialized structure called "Inception," which includes three convolutional layers of different sizes and one max-pooling layer. These layers are used to capture critical features within images. Following this, the model combines the outputs of previous layers to form the final output. To address the issue of overfitting, a sigmoid layer is added, reducing the output layer's dimensionality, which also aids in classification. The model includes a fully connected layer containing hidden units that help recognize more complex patterns in images. In this model, weights are initialized using a Xavier initialization algorithm, enhancing the model's learning capabilities [21]. The architecture and layer structure of the InceptionV3 model are visually detailed in Figure 2.

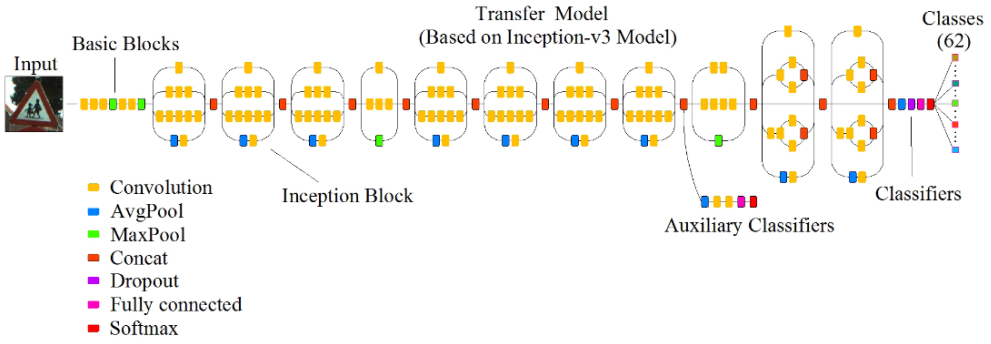

Figure 2. Structure of InceptionV3 [22]

2.2.3. DenseNet169

The DenseNet-169 model, part of the DenseNet family and distinguished by its success in the ImageNet competition, is a deep learning model proposed by G. Huang and colleagues. In this model, each layer receives information from all preceding layers and passes its output to all subsequent layers. This design enhances the model's learning capacity and facilitates improved information flow. DenseNet-169 features an efficient and compact network structure. This compact design reduces the number of parameters and improves computational efficiency. The architecture and layer structure of the DenseNet-169 model are visually detailed in Figure 3 [23].

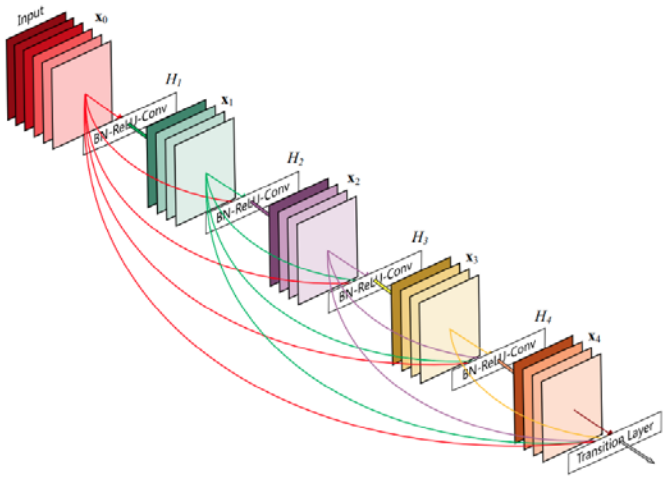

Figure 3. Structure of DenseNet-169 [23]

2.2.4. Bayesian Hyperparameters Optimization

Bayesian optimization is a method used to optimize the performance of machine learning and deep learning models. Hyperparameters are parameters that control the learning process of a model; in neural networks, these typically include values like the number of layers, the number of neurons, and the learning rate. Hyperparameters are often adjusted manually, a time-consuming and inefficient approach. Bayesian optimization offers a faster and more efficient solution for hyperparameter optimization by applying Bayesian optimization principles. The primary objective is to model the hyperparameter space as a probability distribution and update this distribution based on observed performance metrics. By intelligently exploring the hyperparameter search space and focusing on promising regions, Bayesian optimization can identify better performing hyperparameter combinations more efficiently compared to traditional methods. The search space distribution is commonly modeled using a Gaussian Process, which is especially advantageous for complex models with large hyperparameter spaces [24, 25].

The method's operating principle relies on Bayesian optimization techniques and involves core components such as Prior Distribution, Observations, Posterior Distribution, and Acquisition Function. The Prior Distribution represents the probability distribution of possible values in the hyperparameter search space, reflecting prior knowledge about these hyperparameters. In this study, the boundaries of the hyperparameter search space were defined based on a combination of domain knowledge, empirical observations, and computational constraints, aiming to balance search efficiency and model performance. Observations are the performance measurements of models trained with specific hyperparameters, obtained by evaluating the objective function. The Posterior Distribution is an updated probability distribution of hyperparameters based on observations. The Acquisition Function uses the Posterior Distribution to identify the most promising regions in the search space [26]. Figure 4 visually illustrates the behavior and key characteristics of a one-dimensional Gaussian Process.

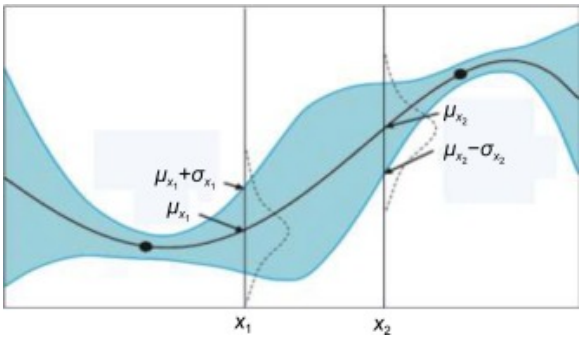

Figure 4. One-dimensional Gaussian Process [25]

2.2.5. Ensemble Learning

Ensemble learning is an approach in machine learning that aims to achieve better results than a single model by combining the predictions of multiple models. This method is based on the idea that the errors of different models will average out, leading to higher accuracy, reduced variance, and improved generalization ability. Ensemble methods can include various models such as decision trees, neural networks, and regression models. The primary advantage of ensemble methods is that, in cases where one model makes an error, other models can compensate for these mistakes, thereby enhancing overall performance. Another significant benefit of ensemble methods is their ability to reduce overfitting. When a single model overfits, it may focus too heavily on specific patterns or noise (irrelevant variance) in the training dataset. However, by combining multiple models, the impact of such errors is minimized, resulting in more generalizable outcomes. Ensemble learning also has some disadvantages. Ensemble models tend to have more complex structures, making them harder to manage and interpret. Additionally, training multiple models and combining their predictions require more computational power and time. Ensemble methods can also have drawbacks related to high resource usage and inter-model dependencies. Generally, four main methods are highlighted when creating an ensemble model: voting, bagging, boosting, and stacking [26, 27].

Voting Ensemble is an ensemble learning method that aims to make an overall prediction by combining multiple classifiers. This approach creates a model that generally outperforms a single classifier by aggregating the predictions of different classifiers, balancing each classifier's weaknesses with the strengths of others to achieve more stable and reliable predictions. There are two primary voting mechanisms: Soft Voting and Hard Voting.

In Soft Voting, classifiers provide a probability distribution for all classes. These probabilities are multiplied by weights reflecting each classifier's strength, where stronger models are assigned, higher weights based on their performance, and then summed to produce the final prediction. The class with the highest total probability is selected as the final prediction. In this method, each classifier's prediction is given a weight corresponding to its confidence level, making the votes of more accurate classifiers more influential.

In Hard Voting, each classifier selects the class it considers most probable, and one vote is cast for that class. The class with the most votes becomes the final prediction. In this method, all classifier votes are treated equally, disregarding probability values [28]. Figure 5 visualizes the hard and soft voting decisions.

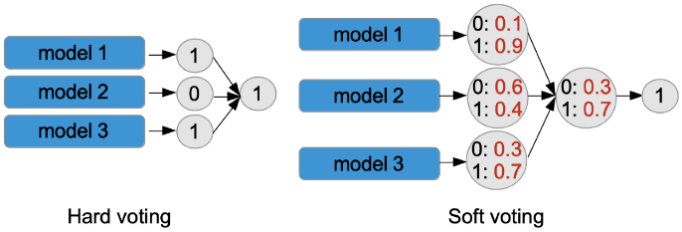

Figure 5. Hard and Soft voting decisions [29]

2.2.6. Performance Evaluation Metrics

A Confusion Matrix is a table used to evaluate the performance of classification algorithms by providing insights into the predicted and actual classes. The rows of the matrix represent the actual classes, while the columns represent the predicted classes. This matrix comprises four fundamental elements [30, 31]:

True Positive (TP): Instances that are actually positive and are correctly predicted as positive by the model.

False Positive (FP): Instances that are actually negative but are incorrectly predicted as positive by the model.

True Negative (TN): Instances that are actually negative and are correctly predicted as negative by the model.

False Negative (FN): Instances that are actually positive but are incorrectly predicted as negative by the model.

Accuracy is a performance evaluation metric calculated by the ratio of the number of correctly predicted instances to the total number of instances. It serves as an indicator of how well the model is performing. Accuracy is particularly useful as a performance measure when the classes are evenly distributed [32]. The method used to calculate the accuracy metric is presented in Equation 2.1.

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN} \tag{2.1}$$

Precision is a performance evaluation metric defined as the ratio of true positive predictions to the total number of positive predictions made by the model. It serves as an indicator of the accuracy of the model's positive predictions. A high precision value indicates that the model has made a minimal number of FP predictions, demonstrating its effectiveness in identifying TP cases [32]. The method used to calculate the precision metric is presented in Equation 2.2.

$$Precision = \frac{TP}{TP+FP} \tag{2.2}$$

Recall is a performance evaluation metric that indicates how effectively the model identifies actual positive instances. It measures the proportion of TP instances that are correctly identified by the model. A high recall value suggests that the model has made few FN predictions, demonstrating its capability

to accurately detect positive cases [32]. The method used to calculate the recall metric is presented in Equation 2.3.

$$Recall = \frac{TP}{TP+FN} \tag{2.3}$$

The *F1 score* is a metric obtained by calculating the harmonic mean of precision and recall performance evaluation metrics. It provides a balance between precision and recall, serving as a more comprehensive measure of performance, particularly in cases where there is an imbalanced class distribution [32]. The method used to calculate the F1-Score metric is presented in Equation 2.4.

$$F1 - Score = 2 * \frac{Precision*Recall}{Precision+Recall} \tag{2.4}$$

3. Results and Discussion

In this study, the ASL dataset obtained from Kaggle was utilized to perform ASL classification using the InceptionV3, DenseNet169, and VGG16 algorithms. Subsequently, the Bayesian optimization algorithm was employed to determine the optimal hyperparameters for the models used. Finally, the InceptionV3, DenseNet169, and VGG16 deep learning algorithms were individually trained on the same dataset to create an Ensemble Learning model using the Soft Voting method. To investigate the impact of hyperparameters on model performance, two different Ensemble Learning models were developed: one utilizing the Bayesian optimization algorithm and one that did not. Their performances were then compared. The dataset was split with 80% allocated for training and 20% for validation, and an early stopping function was added to prevent overfitting. The performance metrics used to measure the training were Accuracy, Precision, Recall, and F1 Score. The validation loss graphs for the model are presented below.

3.1. Results without Hyperparameter Optimization

In this section of the study, the classification performance of deep learning algorithms created without the use of Bayesian optimization, as well as the hyperparameter configurations and results, are presented. The classification performance of the models was measured using performance evaluation metrics such as accuracy, F1 score, and recall. The hyperparameters used in the models were determined manually, and the performance of these hyperparameters was evaluated for comparison with the models optimized using the Bayesian optimization algorithm. The obtained results are visually presented through tables and graphs.

Table 1 presents the hyperparameter and architectural characteristics of the deep learning models created without the use of the Bayesian optimization algorithm. Common hyperparameters across all models include 30 epochs a batch size of 32, Adam optimizer, 2 dense layers, 104 neurons, categorical cross-entropy loss function, a dropout rate of 0.2%, and ReLU-Softmax activation functions. Due to the architectural differences of the deep learning models, the number of layers varies.

Table 1. Parameters of DL algorithms without Bayesian optimization

	Inceptionv3	DenseNet169	VGG16
Layer Size	48	169	16
Model Nodes	5, 5, 2048	7, 7, 1664	7, 7, 512
Epoch	30	30	30
Batch Size	32	32	32
Optimizer	adam	adam	adam
Learning Rate	5e-8	5e-8	2e-7
Dense	2	2	2
# of Neurons	104	104	104
Loss Function	Categorical Crossentropy	Categorical Crossentropy	Categorical Crossentropy
Dropout	0.2	0.2	0.2
Activation Function	ReLU-Softmax	ReLU-Softmax	ReLU-Softmax

Figure 6 shows the training and validation accuracy and loss curves for the InceptionV3 model. These graphs indicate that the model shows steady improvement across epochs. This demonstrates that the model's performance is improving and that it has undergone a successful learning process.

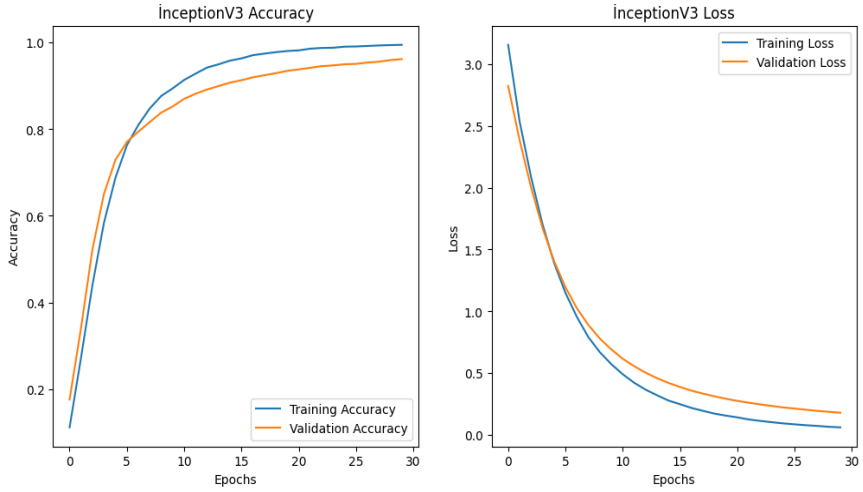

Figure 6. Training and validation accuracy and loss of InceptionV3 model without Bayesian optimization algorithm

Figure 7 presents the training and validation accuracy as well as loss graphs for the DenseNet169 model. These graphs illustrate that the proposed model continues to learn throughout each epoch, with a progressively increasing prediction accuracy. The increase in accuracy and the decrease in loss values during the training process indicate that the model's performance is improving and that it has undergone a successful learning experience.

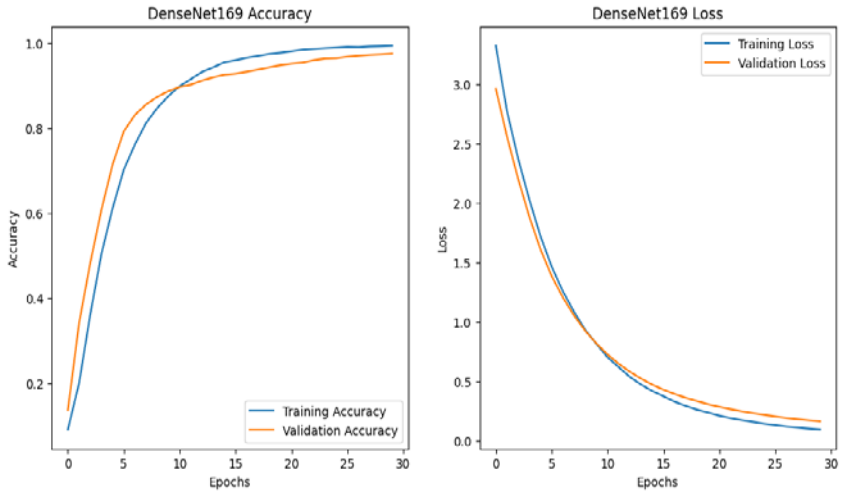

Figure 7. Training and validation accuracy and loss of DenseNet169 model without Bayesian optimization algorithm

Figure 8 exhibits training and validation accuracy along with the loss graphs for the VGG16 model. These graphs indicate that the proposed model continues to learn throughout each epoch, with a gradual increase in prediction accuracy. The rise in accuracy and the decline in loss values during the training process suggest that the model's performance is improving and that it has experienced a successful learning process.

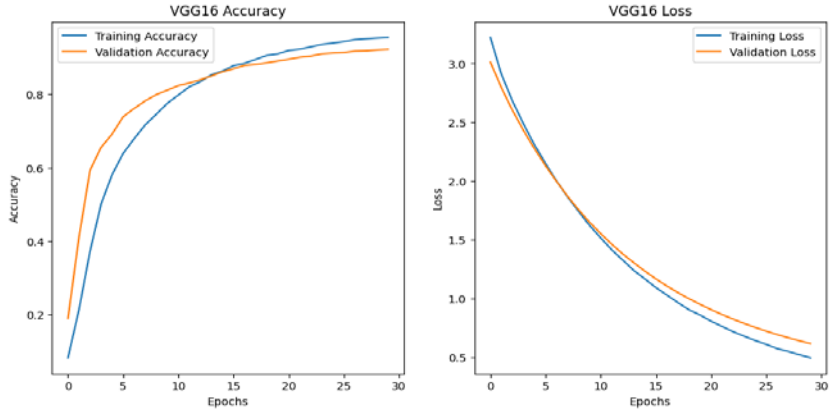

Figure 8. Training and validation accuracy and loss of VGG16 model without Bayesian optimization algorithm

As seen in the outputs of Table 2, the model that demonstrates the best performance on the ASL dataset is the Ensemble Learning model.

Table 2. Performance Comparisons of Deep Learning Algorithm without Bayesian Optimization Algorithm

Method	Accuracy	Precision	Recall	F1-Score
InceptionV3	0.846	0.859	0.846	0.853
DenseNet169	0.883	0.901	0.883	0.892
VGG16	0.789	0.814	0.789	0.802
Ensemble	0.921	0.933	0.921	0.927

3.2. Results with Hyperparameter Optimization

In this section of this study, the classification performance, hyperparameters, and classification results of deep learning algorithms optimized using the Bayesian optimization are presented. The classification performance of the proposed deep learning models was measured using performance evaluation metrics such as accuracy, F1 score, precision, and recall. The configurations of the hyperparameters used in the proposed models and their effects on performance were examined. The obtained results are visually presented through tables and graphs. The outputs in the tables and figures facilitate the observation and interpretation of the impact of hyperparameters on model performance, as assisted by the Bayesian optimization algorithm. The results indicate that the proposed approach can be successfully applied to similar classification problems.

Table 3 presents the hyperparameter and architectural features of the deep learning models created using the Bayesian optimization algorithm. The common hyperparameters across all models include 30 epochs a batch size of 32, the Adam optimizer, 2 dense layers, 104 neurons, the Categorical Crossentropy loss function, and ReLU-Softmax activation functions, which are the same as those used in models without the Bayesian optimization algorithm. Due to architectural differences among the deep learning models, the number of layers varies; additionally, as a result of Bayesian optimization, the learning rate, dropout, and units values differ from model to model. The hyperparameters are specified in Table 3.

Table 3. Parameters of DL algorithms with Bayesian optimization

	Inceptionv3	DenseNet169	VGG16
Layer Size	48	169	16
Model Nodes	5, 5, 2048	7, 7, 1664	7, 7, 512
Epoch	30	30	30
Batch Size	32	32	32
Optimizer	adam	adam	adam
Learning Rate	7e-8 -- 1.5e-7	7e-8 -- 1.5e-7	1e-7 – 6e-7
Dense	2	2	2
# of Neurons	104	104	104
Loss Function	Categorical Crossentropy	Categorical Crossentropy	Categorical Crossentropy
Dropout	0.1 – 0.3	0.1 – 0.3	0.1 – 0.3
Activation Function	ReLU-Softmax	ReLU-Softmax	ReLU-Softmax

Figure 9 presents the training and validation accuracy as well as loss graphs for the InceptionV3 model, where hyperparameters have been optimized using the Bayesian optimization algorithm. These graphs demonstrate that the model continues to learn throughout each epoch and that the prediction accuracy is steadily increasing. The rise in accuracy and the reduction in loss values during the training process indicate that Bayesian optimization has improved the model's performance and facilitated a successful learning process. Furthermore, it has been observed that the Bayesian model outperforms the non-Bayesian approaches.

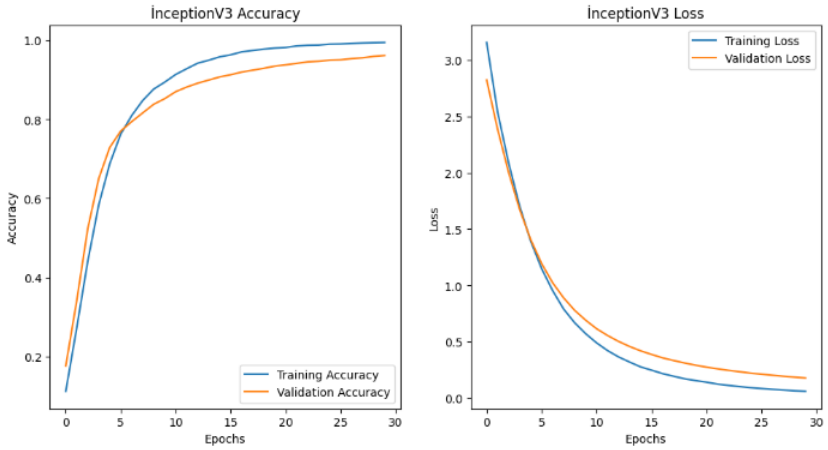

Figure 9. Training and validation accuracy and loss of InceptionV3 model with Bayesian optimization algorithm

Figure 10 presents the training and validation accuracy as well as loss graphs for the DenseNet169 model, where hyperparameters have been optimized using the Bayesian optimization algorithm. These graphs illustrate that the model continues to learn throughout each epoch, with a consistent increase in prediction accuracy. The observed rise in accuracy and decrease in loss values during the training process indicate that Bayesian optimization has improved the model's performance and led to a successful learning process. Additionally, it has been noted that the Bayesian model outperforms the non-Bayesian approaches.

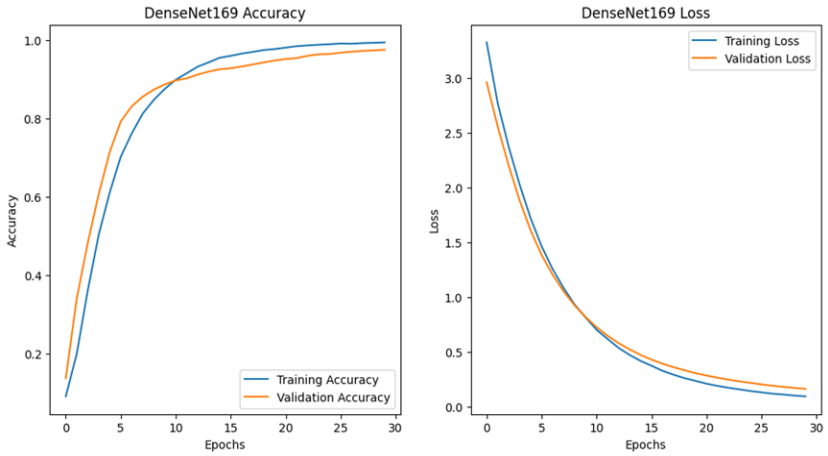

Figure 10. Training and validation accuracy and loss of DenseNet169 model with Bayesian optimization Algorithm

Figure 11 shows the training and validation accuracy as well as loss graphs for the VGG16 model, with hyperparameters optimized using the Bayesian optimization algorithm. These graphs indicate that the model continues to learn through each epoch, demonstrating a steady increase in prediction accuracy. The rise in accuracy and the decrease in loss values throughout the training process suggest that Bayesian optimization has enhanced the model's performance and facilitated a successful learning experience. Additionally, it has been observed that the Bayesian model outperforms non-Bayesian approaches.

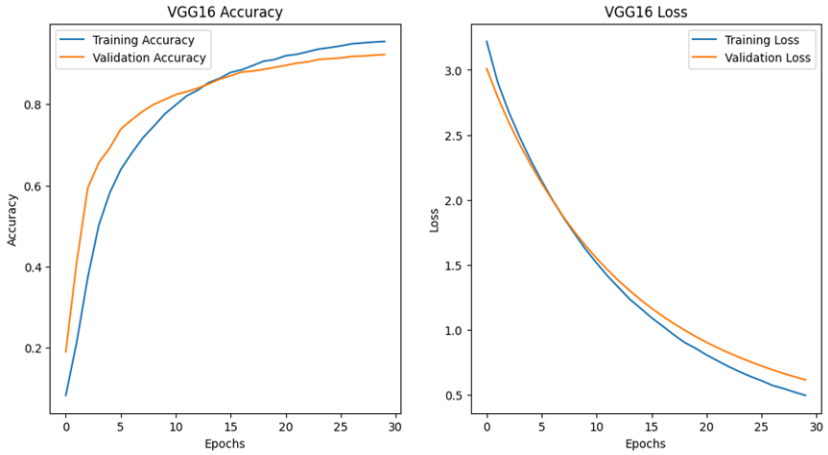

Figure 11. Training and validation accuracy and loss of VGG16 model with Bayesian optimization algorithm

According to the results in Table 4, the best-performing model on the ASL dataset is the Ensemble Learning model with hyperparameters optimized using Bayesian optimization. Furthermore, it has been observed that, in general, Bayesian models outperform non-Bayesian approaches. In this context, the Bayesian Ensemble model achieves the highest success due to both hyperparameter optimization and model combination techniques.

Table 4. Performance Comparisons of Deep Learning Algorithm with Bayesian Optimization Algorithm

Method	Accuracy	Precision	Recall	F1-Score
InceptionV3	0.962	0.963	0.961	0.963
DenseNet169	0.975	0.977	0.975	0.976
VGG16	0.922	0.940	0.922	0.931
Ensemble	0.986	0.987	0.986	0.986

A comparison was made between an Ensemble Learning model created using the Bayesian optimization algorithm and one that did not utilize this algorithm, with the impact of the Bayesian optimization algorithm on model performance illustrated in Figure 11. This comparison highlights the improvements achieved through the optimization process, showcasing the enhancements in accuracy, precision, recall, and F1 score that result from the effective tuning of hyperparameters.

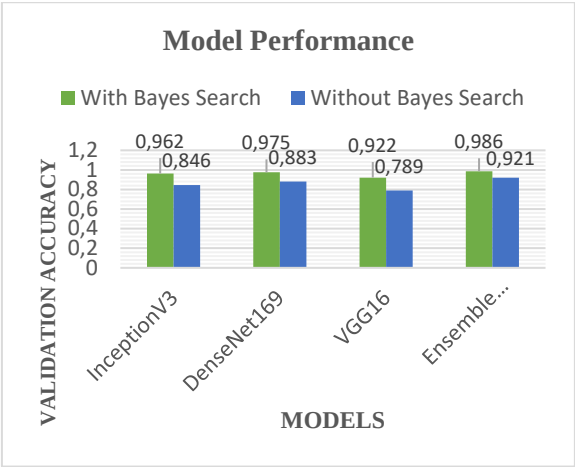

Figure 11. Performance Comparisons of Models

4. Conclusion

The sign language detection application developed in this study offers an effective solution to the communication problems between individuals with hearing impairments and those who do not know sign language. The use of ensemble learning and Bayesian optimization algorithms has significantly enhanced model performance, providing important findings that could guide future research. It is believed that such applications will facilitate the integration of individuals with hearing impairments into society and improve their quality of life. The application leverages artificial intelligence technologies by utilizing image classification models such as InceptionV3, DenseNet169, and VGG16. To achieve a high success rate, these three algorithms have been combined to form an ensemble learning method. In this research, the Bayesian optimization algorithm was employed to examine the impact of hyperparameters on model performance, with comparisons made against other models. According to the results obtained, the ensemble learning model utilizing the Bayesian optimization algorithm demonstrated superior performance compared to other models. This indicates that the Bayesian optimization algorithm positively affects model performance. However, it was also observed that the algorithm has a negative impact on computational costs, particularly in terms of time consumption. This time-related overhead may limit its applicability in real-time or low-latency systems. Therefore, while Bayesian Search offers strong optimization performance, its use in real-time applications should be carefully evaluated, possibly by limiting the search space or applying it during the offline training phase. Nonetheless, it has been identified that the computational cost can be reduced by narrowing the search space. This finding suggests that optimizing the search space should be a focus in future studies to enhance the effectiveness of the Bayesian optimization algorithm. In conclusion, the developed sign language detection application provides an effective solution to the communication problem between individuals with hearing impairments and those who do not know sign language. The implementation of the ensemble learning method and the Bayesian optimization algorithm has played a significant role in improving the model's performance. Future research should concentrate on optimizing the search space of the Bayesian optimization algorithm to reduce computational costs and further enhance model performance. This study is expected to facilitate the integration of individuals with hearing impairments into society and improve their quality of life.

Acknowledgment

The author(s) would like to thank the reviewers and editorial board of the International Journal of Pure and Applied Sciences.

Conflict of Interest

The author(s) declare that there is no conflict of interest in this study. The study was written with the contributions of all authors.

Research and Publication Ethicts Statement

The author(s) declare that they have complied with the scientific, ethical, and citation rules of the International Journal of Pure and Applied Sciences throughout all stages of the study.

References

[1] Quinto-Pozos, D. (2011). Teaching American Sign Language to hearing adult learners. *Annual Review of Applied Linguistics*, *31*, 137-158.

[2] Chowdhary, K., & Chowdhary, K. R. (2020). Natural language processing. *Fundamentals of artificial intelligence*, 603-649. https://doi.org/10.1007/978-81-322-3972-7_19

[3] Abu-Jamie, T. N., & Abu-Naser, S. S. (2022). Classification of sign-language using MobileNet - deep learning. International Journal of Academic Information Systems Research, 6(7), 29–40.

[4] Abu-Jamie, T. N., & Abu-Naser, S. S. (2022). Classification of sign-language using VGG16. *International Journal of Academic Engineering Research*, *6*(6), 36–46.

[5] Murali, R. S. L., Ramayya, L. D., & Santosh, V. A. (2020). Sign language recognition system using convolutional neural network and computer vision. *Int J EngInnov Technol*, 2582-1431.

[6] Pigou, L., Dieleman, S., Kindermans, P. J., & Schrauwen, B. (2015). Sign language recognition using convolutional neural networks. In *Computer Vision-ECCV 2014 Workshops: Zurich, Switzerland, September 6-7 and 12, 2014, Proceedings, Part I 13* (pp. 572-578). Springer International Publishing.

[7] Daroya, R., Peralta, D., & Naval, P. (2018, October). Alphabet sign language image classification using deep learning. In *TENCON 2018-2018 IEEE Region 10 Conference* (pp. 0646-0650). IEEE.

[8] Nareshkumar, M. D., & Jaison, B. (2023). A Light-Weight Deep Learning-Based Architecture for Sign Language Classification. *Intelligent Automation & Soft Computing*, *35*(3).

[9] Bhattacharya, A., Zope, V., Kumbhar, K., Borwankar, P., & Mendes, A. (2019). Classification of sign language gestures using machine learning. *Image*, *8*(12).

[10] Hasan, M. M., Srizon, A. Y., Sayeed, A., & Hasan, M. A. M. (2020, November). Classification of sign language characters by applying a deep convolutional neural network. In *2020 2nd International Conference on Advanced Information and Communication Technology (ICAICT)* (pp. 434-438). IEEE.

[11] Barbhuiya, A. A., Karsh, R. K., & Jain, R. (2021). CNN based feature extraction and classification for sign language. *Multimedia Tools and Applications*, *80*(2), 3051-3069.

[12] Amrutha, K., & Prabu, P. (2021, February). ML based sign language recognition system. In *2021 International Conference on Innovative Trends in Information Technology (ICITIIT)* (pp. 1-6). IEEE.

[13] Abdul, W., Alsulaiman, M., Amin, S. U., Faisal, M., Muhammad, G., Albogamy, F. R., Bencherif, M. A., & Ghaleb, H. (2021). Intelligent real-time Arabic sign language classification using attention-based Inception and BiLSTM. Computers and Electrical Engineering, 95, 107395. https://doi.org/10.1016/j.compeleceng.2021.107395

[14] Leth, P. G. (2023). *Danish sign language recognition in virtual reality using written language ensemble learning* (Master's thesis, Aalborg University).

[15] Kothadiya, D. R., Bhatt, C. M., Rehman, A., Alamri, F. S., & Saba, T. (2023). SignExplainer: an explainable AI-enabled framework for sign language recognition with ensemble learning. *IEEE Access*, *11*, 47410-47419.

[16] Öztürk, Ş., Yiğit, E., & Özkaya, U. (2020). Fused deep features based classification framework for COVID-19 classification with optimized MLP. Konya Journal of Engineering Sciences, 8, 15-27.

[17] Benbakreti, S., Benbakreti, S., & Ozkaya, U. (2024). The classification of eye diseases from fundus images based on CNN and pretrained models.

[18] Bredun, R. (n.d.). *Sign Language: ENG Alphabet* [Data set]. Kaggle. https://www.kaggle.com/datasets/ruslanbredun/sign-language-eng-alphabet

[19] Tammina, S. (2019). Transfer learning using vgg-16 with deep convolutional neural network for classifying images. *International Journal of Scientific and Research Publications (IJSRP)*, *9*(10), 143-150.

[20] Younis, A., Qiang, L., Nyatega, C. O., Adamu, M. J., & Kawuwa, H. B. (2022). Brain tumor analysis using deep learning and VGG-16 ensembling learning approaches. *Applied Sciences*, *12*(14), 7282.

[21] Pan, Y., Liu, J., Cai, Y., Yang, X., Zhang, Z., Long, H., Zhao, K., Yu, X., Zeng, C., Duan, J., Xiao, P., Li, J., Cai, F., Yang, X., & Tan, Z. (2023). Fundus image classification using Inception V3 and ResNet-50 for the early diagnostics of fundus diseases. *Frontiers in Physiology*, *14*, Article 1126780. https://doi.org/10.3389/fphys.2023.1126780

[22] Lin, C., Li, L., Luo, W., Wang, K. C., & Guo, J. (2019). Transfer learning based traffic sign recognition using inception-v3 model. *Periodica Polytechnica Transportation Engineering*, *47*(3), 242-250.

[23] Nair, K., Deshpande, A., Guntuka, R., & Patil, A. (2022). Analysing X-ray images to detect lung diseases using DenseNet-169 technique. Available at SSRN 4111864.

[24] Frazier, P. I. (2018). A tutorial on Bayesian optimization. *arXiv preprint arXiv:1807.02811*.

[25] Wu, J., Chen, X. Y., Zhang, H., Xiong, L. D., Lei, H., & Deng, S. H. (2019). Hyperparameter optimization for machine learning models based on Bayesian optimization. *Journal of Electronic Science and Technology*, *17*(1), 26-40.

[26] Polikar, R. (2012). Ensemble learning. In C. Zhang & Y. Ma (Eds.), *Ensemble Machine Learning: Methods and Applications* (pp. 1–34). Springer US. https://doi.org/10.1007/978-1-4419-9326-7_1

[27] Dietterich, T. G. (2002). Ensemble learning. *The handbook of brain theory and neural networks*, *2*(1), 110-125.

[28] Rojarath, A., Songpan, W., & Pong-inwong, C. (2016, August). Improved ensemble learning for classification techniques based on majority voting. In *2016 7th IEEE international conference on software engineering and service science (ICSESS)* (pp. 107-110). IEEE.

[29] Manconi, A., Armano, G., Gnocchi, M., & Milanesi, L. (2022). A soft-voting ensemble classifier for detecting patients affected by COVID-19. *Applied Sciences*, *12*(15), 7554.

[30] Banda, J. M., Angryk, R. A., & Martens, P. C. H. (2013). Steps toward a large-scale solar image data analysis to differentiate solar phenomena. *Solar Physics*, *288*, 435-462.

[31] Hark, C. (2022). Sahte Haber Tespiti için Derin Bağlamsal Kelime Gömülmeleri ve Sinirsel Ağların Performans Değerlendirmesi. *Fırat Üniversitesi Mühendislik Bilimleri Dergisi*, *34*(2), 733-742.

[32] Powers, D. M. (2020). Evaluation: from precision, recall and F-measure to ROC, informedness, markedness and correlation. *arXiv preprint arXiv:2010.16061*.