Exploring Hand Gesture Classification Using Machine Learning Techniques

Yue Ma
yuema@niu.edu
Northern Illinois University
DeKalb, IL, USA

ABSTRACT

Hand gesture classification is one of the most important processes for hand gesture recognition. Hand gesture classification refers to the task of classifying hand gestures based on their categories. It facilities the application development of augmented reality (AR) tools, especially the ones require precise hand gesture interactions. Current work are restricted on one single model's discussion, with limited metrics for evaluation. In this work, we present the comparisons between 8 classic machine learning models, and 3 CNN models. We evaluate them with various metrics to throughout exploring their performance. In addition to the real hand gesture images, we also consider the AI generated images in this task. Our result shows that pretrained models have superior advantages on image classification tasks. Machine learning models have potentials to facilitate complex tasks such as detecting AI generated images.

1 INTRODUCTION

Hand gesture classification has received significant attention in Human Computer Interaction (HCI)[12, 26, 36], particularly in the augmented reality (AR) field [20, 28, 38]. For example, the HoloLens[8, 27], developed by Microsoft[2], uses the built-in camera to capture human hand gestures as input instead of relying on controllers. To enable smooth interaction, the AR device must be trained with various hand gesture categories, including key gestures such as pointing, grabbing, resizing, and displaying the main menu. Hand gesture classification is also discussed in the context of sign language[4, 19, 30]. While many people are familiar with basic hand gestures such as counting numbers 1, 2, 3, etc., more complex hand gestures are used for communication among specific groups of people with hearing or speaking impairments. Hand gesture classification can facilitate the development of applications that decode the meaning of gestures for people who are not familiar with sign language, allowing them to communicate with those who have speaking and hearing impairments[7]. Most of current work have concentrated on CNN models[16, 25, 31], very less of them presenting classic machine learning solutions. They provided limited evaluation metrics (accuracy and confusion matrices). However there are more evaluation metrics could be explored and compared for this tasks. Moreover, there is very limited work discussing AI generated images in classification tasks, which providing opportunities in further discussion.

The main research question (RQ1) for this paper is to build machine learning models and classify hand gesture images (hand gesture 1, 2, 6, 9 and 14, 5 categories from HG14 datsets[11]) using various classic machine learning models include Stochastic Gradient Descent (SGD), K Nearest Neighbors(KNN), Support Vector Machine(SVM), Random Forest, Decision Tree, ensemble bagging, ensemble boosting Adaboost, gradient boosting. We will also implement with Convolutional Neural Networks (CNN), which are often discussed for image processing tasks. Personal customized CNN architecture and several pretrained models are compared. Accuracy, precision, recall, various metrics will be used for model evaluation. Recently, AI generated models got a lot of attention, especially chatGPT which is widely discussed nowadays. Inspire by this, we further extend a hand gesture category with AI generated hand gesture images (obtained from online AI image generator Craiyon [1]) in order to test and compare model performance with noises. This will be our extended research question (RQ2).

Contributions of this work is that we perform the hand gesture classification with various models that are applicable to image classification tasks, and we also provide throughout evaluation metrics that help to evaluate the performance of various models. The additional task considers AI generated images. This process could further facilitate model to have a better understanding of the characteristics of human's hand, which could also reflect the limitations of the website that we are using to collect the AI generated images.

2 RELATED WORK

Hand gesture recognition which contains both tracking, locating process as long as the hand gesture classification process, so papers regarding hand gesture recognition will be included in related work as well. Most of works which are discussing hand gesture classification are using convolutional neural networks[3, 9, 35]. Gadekallu et al. [10] implements CNN with a crow search algorithm (CSA). The model is compared with other CNN models that implements with other algorithms, however, their evaluation metrics only contain accuracy and loss. Similarly, Jo et al. [13] uses a combined structure Convolutional Recurrent Neural Network (CRNN) (Other combined model examples [5, 18, 29, 34]) for real time hand gesture classifications. Their work utilizes the advantages of RNN for time series data processing as well as CNN for image feature extractions. Accuracy, error, time delayed and confusion matrix are used for evaluation. Without providing these information, it is less confident to trust model's overall results. Mendes et al. [23] uses ANN (Artificial Neural Network) for classify static hand gestures and HMM (Hidden Markov models) to classify dynamic hand gestures. In their work only confusion matrix is used for comparisons. However, there are more important metrics that could be used to evaluate the model, such as precision, recall, F-1 score, etc. Trigo et al. [32] use simple multi-layer perceptron, instead of using images as the direct input to the model, their work compares the model performance based on different set of features generated by various image descriptors(Invariant Moments, K-curvature and template matching), and propose a Geometric Shape Descriptors. Their work

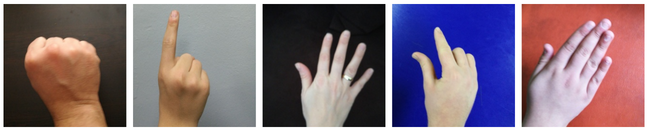

Figure 1: Selected hand gestures from HG 14 dataset. From left to right: Hand Gesture 1, 2, 6, 9 and 14.

Figure 2: AI generated hand gesture image examples collected from Craiyon website. 500 images with random gestures are viewed as one additional category added to the original dataset.

only uses 'Correctly Classified Instances' as the evaluation metrics, which is restricted to have a general view of the entire model.

In addition to neural networks, K nearest neighbors method is used for the classification process in the work of Kollorz et al. [17]. The projections of hand on x, y axes together with depth features (e.g., maximum/minimum projection values) are used for hand gesture classification. In this work, the training data is not clear, and only confusion matrix is used for evaluation. In conclusion, there are many work considered how to classify different hand gestures or other subjects, however, there is less work took AI generated images into considerations when building classifiers.

3 METHODOLOGY

3.1 Data Collection

Dataset used for RQ1: HG14 [11] contains 14 various hand gestures, and each gesture contains 1000 images with different background. These hand gestures are collected from 17 different people. The size of these images are 256*256, with 3 colored channel (RGB). The HG14 dataset is specifically collected for hand gesture recognition of augmented reality application. Machine learning models require a longer time when training on images compared with other types of data (e.g., text, numerical values), due to the heavy training time consumed and the time limitation of this project, only part of hand gestures will be selected (hand gesture 1, 2, 6, 9, 14) for the hand gesture classification task and only 500 images will be selected for each category (Figure 1).

Dataset used for RQ2: We add extra one class which contains 500 AI generated hand gesture images, which is manually collected from an online AI image generator Craiyon [1], to the previous hand gesture dataset for RQ1. Figure 2 shows some example images. Craiyon takes text from user as input, and it can return 9 images per batch for the given description. There are several reasons for choosing Crayion generator instead of other generator. Firstly, the website is completely free to use, with no restrictions on the number of times a user can generate images. Every batch of generated images are unique from other batches. Second, the generated hand gesture are mostly at the center of image, which is in consistent with the images from the HG14 dataset.

3.2 Data Preprocessing

3.2.1 Classic machine learning models data preprocessing. The size of original images are 256*256 which indicates that there are 256 pixels in height and 256 pixels in width, with 3 color channel. Without resizing the images, it will generated 256*256*3 = 196,608 features. Hence, we will resize the images to 150* 150 in order to reduce the computation load for these classic machine learning models. We will also flatten the data by converting the 3 dimension color image (2 dimension when discussed spatial dimensions in height and width) to a flattened one dimensional array. By doing these above steps, the features for training are reduced to 150*150*3=67,500, but it still needs more preprocessing steps. Too many feature inputs will lead to overfitting issues [37] in several machine learning models. Feature reduction method for example Principal Component Analysis (PCA) is considered in this case[15]. PCA is a linear unsupervised feature reduction method [33], which is especially effective when reducing features that are correlated[24]. Image data can be highly correlated [22]. We used PCA to transform our pixel features to a new set of features which is called principle components. In this project, the principle components is set to 200. It is greatly reduced the feature complexity of the original data. The final shape of input x after preprocessed is (2500,200), since there are 5 category images and each category contains 500 images, 500*5 is the first dimension. The second dimension is the preset 200 principle components. The final shape of input y is (2500,), one image has a label category, so there are 2500 labels corresponding each image. Next, we split the data, 85% will be used for training, and 15% will be used for testing. For the second task R2, we simply keep the same preprocessed steps while add one extra AI generated images category when processing the input data.

3.2.2 Convolutional Neural Networks data preprocessing. For convenient of programming, compared to images for classic machine learning models which put each category of hand gesture images in different folders then split the training, testing data, for CNN models we pre-split images to training, validation and testing folders. In each folder, there are several sub-folders contains different category of hand gestures. For R1, there are 1750 (70%) images for training, 375 (15%) images for validation, and 375 (15%) images for testing. All the images are resized to 224*224, since several pretrained models are used for comparisons, and these models require size 224*224 for all the image inputs. The preprocessed steps for CNN are less complicated than classic machine learning models, since the best part of using CNN for image processing is that it could keep the spatial dimensions during training instead of directly flattening them to extract features. It is good to keep spatial dimensions since it reflects the relationships between pixels, it keeps all the important features, and it will also benefit to other higher level image classification tasks, for example, semantic segmentation which classify pixels based on the category it belongs to.

3.3 Model Setup

For the SGD classifier, we uses logistic loss function, shuffle is set to be true. For SVM, we tested with 3 kernel functions (sigmoid,

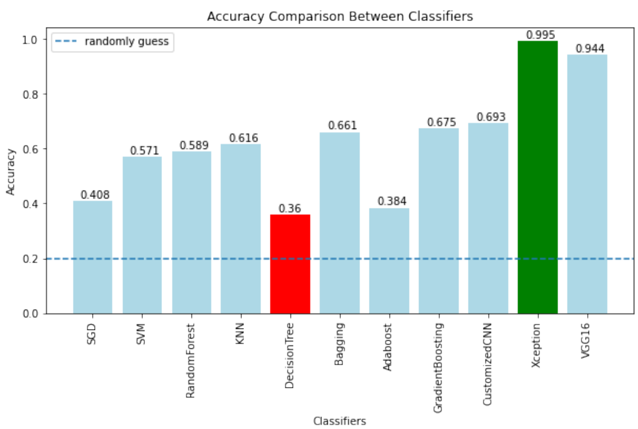

Figure 3: Accuracy Comparison

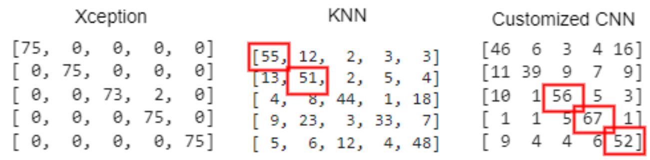

Figure 4: Confusion Matrix of example models

rbf and polynomial), since polynomial has the best performance during testing, we selected polynomial as the final kernel function, with gamma value 0.5, probability is true. 100 decision trees are used for random forest classifier. For KNN, we specify 5 as the number of neighbors, since we have 5 categories for RQ1, this value is changed to 6 for RQ2. Gini criterion is used for decision tree. Bagging method and Adaboost, Gradient Bootsing all use 100 decision tree classifiers as the based estimator to compare the performance with the performance of random forest. There are 4 convolutional layers, 3 max pooling layers for CNN model, and the flattened result is sent to fully connected network with 2 dense layer and one dropout layer.

4 RESULT

4.1 RQ1 Result

4.1.1 Accuracy and Loss. Accuracy is the ratio of correctly predicted instances to the total number of instances. It is a common measurement for evaluating classification performance, which provides an overview of the classifiers' performance. Considering training accuracy, some of the classic machine learning models has overfitting issues, SVM, Random Forest, bagging and boosting methods (Adaboost and gradient boosting) the training accuracy is closed to 1.00, while the testing accuracy is around 0.5-0.7. There are couple reasons, first even though we cut the size of features passing to the network, there are still 200 features sent to the network. Second, it is because of the native structure of methods, for example, bagging and boosting methods are prone to overfitting[21] due to the weighted voting setup. For all the CNN models, we also checked the accuracy and loss evolution during training epochs. Both Xception and VGG16 pretrained models reach a higher accuracy, lower loss at an early stage, while our customized CNN shows steady increasing trend during training for accuracy and steady decreasing trend for loss evolution graphs. The VGG16 and our customized CNN has some overfitting issues, since we observed that the validation set has a comparatively lower accuracy and higher loss than the training set. However, it is not as serious as the classic machine learning models, which shows a large gap for the performance on seen and unseen datasets.

Figure 3 shows the accuracy comparison between all the models using the test dataset. All models are useful to a certain extent because they all perform better than the baseline, which is typically a random guess. For the classic machine learning models, the gradient boosting model has the best performance, which reached 0.675 accuracy. However, the decision tree model shows the poor result, which only has 0.36 accuracy. For CNN models, our customized model has an accuracy of 0.693, outperforming all classic machine learning models, but yielding poorer results compared to other pretrained models. For pretrained models, Xception has the best result reaching 0.995 accuracy during testing.

4.1.2 Classification Report. Precision and recall both prefer a larger value, as closer to 1 will be better. But they require a harmony that a classifier has a better performance when the precision and recall both have a higher value. High precision and low recall shows the classifier returns less results but most of them are correct. While low precision and high recall shows that the classifier returns many results but less of them are correct. To achieve the harmony between precision and recall, we have the F-1 score. F-1 score is calculated based on the value of precision and recall. It achieves the balance between the two important indicators, that is the precision and recall, and hence it is a suitable indicator for measuring and evaluating the imbalanced dataset[14]. First, observing from Table 1, we can find that since our dataset is a balanced one, for all the models there are no cases that are worth our awareness that one value is high and the other value is significantly low. Without comparing with pretrained models, and based on F-1 score KNN and gradient boosting have the best performance on classifying hand gesture 1 (label 0), gradient boosting also has the best performance on classifying hand gesture 2 (label 1) and 9 (label 3). Our customized CNN has the best performance on classifying hand gesture 6 (label 2)and 14 (label 4). In conclusion, gradient boosting has the best performance in regards to the classification report, even though its testing set accuracy is lower than the customized CNN.

4.1.3 Confusion Matrix. Confusion matrix provides the raw information [6] of supervised classification results for a dataset. It compares between the actual classes and the predicted classes, and it is often used for the test set. Figure 4 shows the confusion matrix of some example models in the project. Among two pretrained models, Xception has the best performance. It correctly classifies all the classes except 2 mistakes for class 2 (hand gesture 6). Without considering pretrained models, KNN shows its advantage on classifying the images from the first 2 classes. Customized CNN correctly classifies most of the images from the other 3 classes compared to other models.

Table 1: Summary of classification reports of classifiers

	Label	SGD	SVM	RF	KNN	DT	Bagging	Adaboost	GB	C-CNN	Xception	VGG16
precision	**0**	0.33	0.51	0.58	0.64	0.34	0.69	0.45	0.77	0.60	1.00	0.93
	1	0.53	0.64	0.62	0.51	0.43	0.58	0.37	0.70	0.76	1.00	0.97
	2	0.35	0.67	0.63	0.70	0.44	0.69	0.42	0.74	0.75	0.97	0.99
	3	0.48	0.56	0.58	0.72	0.32	0.63	0.33	0.65	0.64	1.00	1.00
	4	0.39	0.52	0.54	0.60	0.31	0.74	0.36	0.58	0.73	1.00	0.85
recall	**0**	0.35	0.57	0.60	0.73	0.27	0.61	0.44	0.61	0.61	1.00	0.95
	1	0.36	0.57	0.61	0.68	0.51	0.67	0.35	0.64	0.52	1.00	0.79
	2	0.31	0.56	0.59	0.68	0.28	0.68	0.39	0.64	0.89	1.00	0.99
	3	0.53	0.52	0.56	0.44	0.31	0.67	0.33	0.80	0.69	1.00	1.00
	4	0.49	0.63	0.59	0.64	0.44	0.68	0.41	0.68	0.75	0.97	1.00
F1-score	**0**	0.34	0.54	0.59	0.68	0.30	0.65	0.45	0.68	0.61	1.00	0.94
	1	0.43	0.61	0.62	0.58	0.46	0.62	0.36	0.67	0.62	1.00	0.87
	2	0.33	0.61	0.61	0.64	0.34	0.68	0.40	0.69	0.82	0.99	0.99
	3	0.50	0.54	0.57	0.55	0.31	0.65	0.33	0.71	0.67	1.00	1.00
	4	0.44	0.57	0.56	0.62	0.36	0.71	0.38	0.63	0.74	0.99	0.92

4.2 RQ2 Result

Table 2 shows the confusion matrix of classifiers on classifying AI generated images, we can find that most models have a great performance on correctly classifying the AI generated images from the website Craiyon. Customized CNN model and VGG16 have bad performance on this task. They falsely classify some AI generated images into hand gesture 6 (label 2), 9 (label 3) and 14 (label 4). Compare to classifying other real hand gesture images, most models has the best performance on classifying AI generated images. It shows the machine learning models have the ability to deal with complex tasks, however it also implies the limitation of the AI image model used by the website Craiyon. Some of the hand gestures are very close to the real person's hand. However, most images are still able to tell by human's eyes, since the texture of skin is slightly different with a normal human's hand which makes them suspicious. Certain images are easier to discern than others, since they have a cartoon style, which is apparently different with real hands, and some of them have extra fingers which are different with 5 fingers of human in normal case, and also some of them have a strange hand gesture, which is almost impossible for human to do. AI image generator, it requires tons of training images of each object in order to provide correct output. Craiyon website did not provide too much detailed information about their training data for the model, so we would suggest that they still need more training images for their model to provide a higher quality AI generated images. We also observed that with the joining of AI generated images, the overall accuracy for most of the models are higher than the performance without this class. This may highly due to the high accuracy on classifying AI generated image class. Precision and recall value for real hand gesture classes are slightly increased or decreased. We could not draw the conclusion that the join of AI generated images will facilitate the pattern learning of other real hand gestures.

Table 2: Confusion Matrix of classifiers on classifying AI generated images

		Actual					
	Class	**0**	**1**	**2**	**3**	**4**	**AI**
Predicted	**SGD**	0	0	0	0	1	74
	SVM	0	0	0	0	0	75
	RF	0	0	0	0	0	75
	KNN	0	0	0	0	0	75
	DT	0	0	0	0	0	75
	Bagging	0	0	0	0	0	75
	AdaBoost	0	0	0	0	0	75
	GB	0	0	0	0	0	75
	C-CNN	0	0	1	1	4	69
	Xception	0	0	0	0	0	75
	VGG16	0	0	2	3	1	70

5 CONCLUSION

This work explores hand gesture classification tasks with various machine learning methods, including classic machine learning models, Customized CNN models, and two pertained models Xception and VGG16. We used various evaluation metrics for this tasks. In conclusion, pertained models show their great advantages on classification tasks when the size of dataset is limited. Without considering pertained models, our customized CNN, KNN, and gradient boosting models have the better result than other models. Our result shows that machine learning models have the potential to differentiate between real images and AI generated images, which also shows the limitation of the AI image generator that we used to generate the images. The limitation of this work is that we only selected limited categories of images. For future work, we can increase the size of the input, incorporate more hand gesture categories, and we can also try with more AI image generators.

REFERENCES

[1] 2023. Craiyon. https://www.craiyon.com/

[2] 2023. Microsoft HoloLens2. https://www.microsoft.com/en-us/hololens

[3] Ali A Alani, Georgina Cosma, Aboozar Taherkhani, and TM McGinnity. 2018. Hand gesture recognition using an adapted convolutional neural network with data augmentation. In *2018 4th International conference on information management (ICIM)*. IEEE, 5–12. https://doi.org/10.1109/INFOMAN.2018.8392660

[4] Abul Abbas Barbhuiya, Ram Kumar Karsh, and Rahul Jain. 2021. CNN based feature extraction and classification for sign language. *Multimedia Tools and Applications* 80, 2 (2021), 3051–3069. https://doi.org/10.1007/s11042-020-09829-y

[5] Nanziba Basnin, Lutfun Nahar, and Mohammad Shahada Hossain. 2021. An integrated CNN-LSTM model for micro hand gesture recognition. In *Intelligent Computing and Optimization: Proceedings of the 3rd International Conference on Intelligent Computing and Optimization 2020 (ICO 2020)*. Springer, 379–392. https://doi.org/10.1007/978-3-030-68154-8_35

[6] Olivier Caelen. 2017. A Bayesian interpretation of the confusion matrix. *Annals of Mathematics and Artificial Intelligence* 81, 3-4 (2017), 429–450. https://doi.org/10.1007/s10472-017-9564-8

[7] Anna Cavender and Richard E Ladner. 2008. Hearing impairments. *Web accessibility: A foundation for research* (2008), 25–35. https://doi.org/10.1007/978-1-84800-050-6_3

[8] Nikolas Chaconas and Tobias Höllerer. 2018. An evaluation of bimanual gestures on the microsoft hololens. In *2018 IEEE conference on virtual reality and 3D user interfaces (VR)*. IEEE, 1–8. https://doi.org/10.1109/VR.2018.8446320

[9] Zhaoxi Chen, Gang Li, Francesco Fioranelli, and Hugh Griffiths. 2019. Dynamic hand gesture classification based on multistatic radar micro-Doppler signatures using convolutional neural network. In *2019 IEEE Radar Conference (RadarConf)*. IEEE, 1–5. https://doi.org/10.1109/RADAR.2019.8835796

[10] Thippa Reddy Gadekallu, Mamoun Alazab, Rajesh Kaluri, Praveen Kumar Reddy Maddikunta, Sweta Bhattacharya, and Kuruva Lakshmanna. 2021. Hand gesture classification using a novel CNN-crow search algorithm. *Complex & Intelligent Systems* 7 (2021), 1855–1868. https://doi.org/10.1007/s40747-021-00324-x

[11] Osman Güler and İbrahim Yücedağ. 2022. Hand gesture recognition from 2D images by using convolutional capsule neural networks. *Arabian Journal for Science and Engineering* 47, 2 (2022), 1211–1225. https://doi.org/10.1007/s13369-021-05867-2

[12] Aashni Haria, Archanasri Subramanian, Nivedhitha Asokkumar, Shristi Poddar, and Jyothi S Nayak. 2017. Hand gesture recognition for human computer interaction. *Procedia computer science* 115 (2017), 367–374. https://doi.org/10.1016/j.procs.2017.09.092

[13] Yong-Un Jo and Do-Chang Oh. 2020. Real-time hand gesture classification using crnn with scale average wavelet transform. *Journal of Mechanics in Medicine and Biology* 20, 10 (2020), 2040028. https://doi.org/10.1142/S021951942040028X

[14] Firuz Kamalov, Fadi Thabtah, and Ho Hon Leung. 2022. Feature selection in imbalanced data. *Annals of Data Science* (2022), 1–15. https://doi.org/10.1007/s40745-021-00366-5

[15] Julius M Kernbach and Victor E Staartjes. 2022. Foundations of machine learning-based clinical prediction modeling: Part II—Generalization and overfitting. *Machine Learning in Clinical Neuroscience: Foundations and Applications* (2022), 15–21. https://doi.org/10.1007/978-3-030-85292-4_3

[16] Youngwook Kim and Brian Toomajian. 2016. Hand gesture recognition using micro-Doppler signatures with convolutional neural network. *IEEE Access* 4 (2016), 7125–7130. https://doi.org/10.1109/ACCESS.2016.2617282

[17] Eva Kollorz, Jochen Penne, Joachim Hornegger, and Alexander Barke. 2008. Gesture recognition with a time-of-flight camera. *International Journal of Intelligent Systems Technologies and Applications* 5, 3-4 (2008), 334–343. https://doi.org/10.1504/IJISTA.2008.021296

[18] Kenneth Lai and Svetlana N Yanushkevich. 2018. CNN+ RNN depth and skeleton based dynamic hand gesture recognition. In *2018 24th international conference on pattern recognition (ICPR)*. IEEE, 3451–3456. https://doi.org/10.1109/ICPR.2018.8545718

[19] Jeroen F Lichtenauer, Emile A Hendriks, and Marcel JT Reinders. 2008. Sign language recognition by combining statistical DTW and independent classification. *IEEE transactions on pattern analysis and machine intelligence* 30, 11 (2008), 2040–2046. https://doi.org/10.1109/TPAMI.2008.123

[20] Attila Licsár and Tamás Szirányi. 2004. Dynamic training of hand gesture recognition system. In *Proceedings of the 17th International Conference on Pattern Recognition, 2004. ICPR 2004.*, Vol. 4. IEEE, 971–974. https://doi.org/10.1109/ICPR.2004.1333935

[21] Richard Maclin and David Opitz. 1997. An empirical evaluation of bagging and boosting. *AAAI/IAAI* 1997 (1997), 546–551.

[22] Angshul Majumdar and Rabab K Ward. 2011. Joint reconstruction of multiecho MR images using correlated sparsity. *Magnetic resonance imaging* 29, 7 (2011), 899–906. https://doi.org/10.1016/j.mri.2011.03.008

[23] Nuno Mendes, João Ferrer, João Vitorino, Mohammad Safeea, and Pedro Neto. 2017. Human behavior and hand gesture classification for smart human-robot interaction. *Procedia Manufacturing* 11 (2017), 91–98. https://doi.org/10.1016/j.promfg.2017.07.156

[24] MZF Nasution, OS Sitompul, and M Ramli. 2018. PCA based feature reduction to improve the accuracy of decision tree c4. 5 classification. In *Journal of Physics: Conference Series*, Vol. 978. IOP Publishing, 012058. https://doi.org/10.1088/1742-6596/978/1/012058

[25] Mehmet Akif Ozdemir, Deniz Hande Kisa, Onan Guren, and Aydin Akan. 2022. Hand gesture classification using time–frequency images and transfer learning based on cnn. *Biomedical Signal Processing and Control* 77 (2022), 103787. https://doi.org/10.1016/j.bspc.2022.103787

[26] Meenakshi Panwar and Pawan Singh Mehra. 2011. Hand gesture recognition for human computer interaction. In *2011 International Conference on Image Information Processing*. IEEE, 1–7. https://doi.org/10.1109/ICIIP.2011.6108940

[27] Philip Pratt, Matthew Ives, Graham Lawton, Jonathan Simmons, Nasko Radev, Liana Spyropoulou, and Dimitri Amiras. 2018. Through the HoloLens™ looking glass: augmented reality for extremity reconstruction surgery using 3D vascular models with perforating vessels. *European radiology experimental* 2 (2018), 1–7. https://doi.org/10.1186/s41747-017-0033-2

[28] Stefan Reifinger, Frank Wallhoff, Markus Ablassmeier, Tony Poitschke, and Gerhard Rigoll. 2007. Static and dynamic hand-gesture recognition for augmented reality applications. In *Human-Computer Interaction. HCI Intelligent Multimodal Interaction Environments: 12th International Conference, HCI International 2007, Beijing, China, July 22-27, 2007, Proceedings, Part III 12*. Springer, 728–737. https://doi.org/10.1007/978-3-540-73110-8_79

[29] Sungho Shin and Wonyong Sung. 2016. Dynamic hand gesture recognition for wearable devices with low complexity recurrent neural networks. In *2016 IEEE International Symposium on Circuits and Systems (ISCAS)*. IEEE, 2274–2277. https://doi.org/10.1109/ISCAS.2016.7539037

[30] Joyeeta Singha and Karen Das. 2013. Indian sign language recognition using eigen value weighted euclidean distance based classification technique. *arXiv preprint arXiv:1303.0634* (2013). https://doi.org/10.48550/arXiv.1303.0634

[31] Josiah W Smith, Shiva Thiagarajan, Richard Willis, Yiorgos Makris, and Murat Torlak. 2021. Improved static hand gesture classification on deep convolutional neural networks using novel sterile training technique. *Ieee Access* 9 (2021), 10893–10902. https://doi.org/10.1109/ACCESS.2021.3051454

[32] Thiago R Trigo and Sergio Roberto M Pellegrino. 2010. An analysis of features for hand-gesture classification. In *17th international conference on systems, signals and image processing (IWSSIP 2010)*. 412–415.

[33] Md Palash Uddin, Md Al Mamun, and Md Ali Hossain. 2021. PCA-based feature reduction for hyperspectral remote sensing image classification. *IETE Technical Review* 38, 4 (2021), 377–396. https://doi.org/10.1080/02564602.2020.1740615

[34] Muneeb Ur Rehman, Fawad Ahmed, Muhammad Attique Khan, Usman Tariq, Faisal Abdulaziz Alfouzan, Nouf M Alzahrani, and Jawad Ahmad. 2021. Dynamic hand gesture recognition using 3d-cnn and lstm networks. *Computers, Materials & Continua* 70, 3 (2021). https://doi.org/10.32604/cmc.2022.019586

[35] Kollipara Sai Varun, I Puneeth, and T Prem Jacob. 2019. Hand gesture recognition and implementation for disables using CNN'S. In *2019 International Conference on Communication and Signal Processing (ICCSP)*. IEEE, 0592–0595. https://doi.org/10.1109/ICCSP.2019.8697980

[36] Mais Yasen and Shaidah Jusoh. 2019. A systematic review on hand gesture recognition techniques, challenges and applications. *PeerJ Computer Science* 5 (2019), e218. https://doi.org/10.7717/peerj-cs.218

[37] Xue Ying. 2019. An overview of overfitting and its solutions. In *Journal of physics: Conference series*, Vol. 1168. IOP Publishing, 022022. https://doi.org/10.1088/1742-6596/1168/2/022022

[38] Jong-Hyun Yoon, Jong-Seung Park, and Mee Young Sung. 2006. Vision-Based bare-hand gesture interface for interactive augmented reality applications. In *Entertainment Computing-ICEC 2006: 5th International Conference, Cambridge, UK, September 20-22, 2006. Proceedings 5*. Springer, 386–389. https://doi.org/10.1007/11872320_56