JETIR.ORG ISSN: 2349-5162 | ESTD Year : 2014 | Monthly Issue
JOURNAL OF EMERGING TECHNOLOGIES AND INNOVATIVE RESEARCH (JETIR)
An International Scholarly Open Access, Peer-reviewed, Refereed Journal

EXPLORING MACHINE LEARNING FOR REAL-TIME GESTURE ANALYSIS

Mr. Gowtham S[1]
Sivadharani S[2], Dhanush Chandar N A[3], Elizabeth R[4],
[1]Assistant Professor, Department of Information Technology,
K.S.R.College of Engineering, (Autonomous)
[2,3,4] B.Tech., Department of Information Technology, K.S.R. College of Engineering, (Autonomous)

Abstract:

Many people utilize gesture recognition to engage with technology and others, as well as to express their feelings. Given that hand gesture identification is a high-dimensional pattern recognition challenge, researchers are highly interested in it. The performance of machine learning models is directly impacted by the enormous dimensionality of the problem. Feature extraction and selection can be used to solve the dimensionality issue. In order to do this, a model evaluation that makes use of both human and automatic feature extraction was suggested. While a CNN and BiLSTM were used for the automatic feature extraction, the statistical functions of central tendency were used for the manual feature extraction. Additionally, SVM, ANN, and Softmax classifiers assessed these properties.

Keywords: Convolutional Neural Network, Feature Extraction, Hand Gesture Recognition, Leap Motion Controller.

1. Introduction:

There exists a strong correlation between the quantity and quality of data and the efficacy and generalization of machine learning models. The amount of data collected depends on the nature of the problem, the technology employed for data collection, and the accessibility of the data. Generally speaking, machine learning models perform better on larger data sets.

The lack of bias, noise, and errors in the data is referred to as data quality, and it can improve the machine learning models' accuracy and reliability. The model requires additional data to have a better chance of detecting the underlying patterns and correlations in the data, which will enhance predictions and generalization with new, unknown data. Additionally, the quality of the data is determined by how well the features.

In this way, the issue of dimensionality and machine learning models are tightly connected. When machine learning models have a lot of features, this problem arises. Even in scenarios where an overfitting problem is almost a given, machine learning models can nevertheless work Overfitting happens when there are too many features compared to the quantity of training data and not enough data. Adding an excessive number of features might also make the model more computationally demanding, which makes it harder to train and apply in real- world scenarios.

The Techniques for reducing dimensionality must be used in this situation. Feature selection and

feature extraction are the techniques related to dimensionality reduction. The process of choosing the optimal functions to represent the problem is known as feature selection. However, the process of choosing and altering a dataset's most pertinent and instructive elements is known as feature extraction. An essential phase in machine learning models is feature extraction . This is due to the fact that the features' quality and applicability have a significant impact on the model's performance and accuracy. Due to the initial data frequently comprising a large number of features that are superfluous, noisy, or unrelated to the task at hand, overfitting and poor generalization performance might occur.

To prevent these issues, it is crucial to carefully choose and preprocess the most essential and instructive aspects for a particular activity. To determine which attributes are most significant and to eliminate those that are unnecessary or redundant, this may require domain expertise, statistical analysis, and data visualization approaches. The method of getting the greatest features can be approached in two ways: automatically through feature extraction or manually through feature extraction.

Automatic feature extraction is one machine learning technique that makes use of deep learning and the Recurrent neural networks (RNN) or CNN could be utilized to automatically extract features for hand gesture identification using the input from the Leap Motion Controller. Furthermore, computerized feature extraction might possibly find more complex and informative features than manual approaches for hand gesture detection, while also saving time and effort in feature engineering. It might be more challenging to analyze and comprehend than manual approaches, though, and it needs a lot of processing power and training data.

Signal processing and feature engineering expertise as well as domain knowledge are needed for manual feature extraction for hand gesture detection. Finding the most informative features for a particularapplication can be labor- and time- intensive, and it may need repeated testing and improvement. But because the characteristics are intentionally defined andchosen using human understanding and intuition, it can also be easier to digest and comprehend than automatic featureextraction techniques.

2. Related Work:

The input data should be spatial locations as time series acquired via the Leap Motion Controller. Furthermore, the data should be analyzed using classifiers such as CNN-ANN, CNN-SVM, BiLSTM-ANN, and BiLSTM-SVM. They suggest atime series consisting of a set of photos. Additionally, because LSTM is a recurrent neural network that operates with time series and CNN extracts features, they suggest combining the two techniques. Since we suggest analysing features in a hand gesture recognition system in this research, it is also vital to clarify the distinction between classification and recognition. An observation is sent to the classification problem, the classification algorithm generates a label, and this label is subsequently mapped to a list of prepared labels in order to assign an observation to the appropriate class. The algorithm's need to be able to identify an observation into a predicted class and pinpoint the precise instant the motion is made is what makes recognition challenging. The second half of the study is structured as follows: the technique section includes an overview of the work, dataset construction, machine learning algorithms, CNN, and BiLSTM; the experimentation and outcomes section follows; and lastly, conclusions and a presentation are given. The authors suggested dynamic hand gesture recognition (DHGR) using long short-term memory (LSTM) and convolutional neural networks (CNN). Leap Motion Controller data was collected. They make use of a dataset made up of information that was taken straight out of the Leap Motion Controller. Eight gestures are included in the LMDI dataset: expand, grasp, pinch, tap, clockwise, counterclockwise, swipe left, and swipe right. The authors suggest a hand gesture using data from the Leap Motion Controller.

It follows that feature extraction is carried out manually. The characteristics consisted of five normalized lengths between the palm center and each fingertip. They show how to use CNN and camera data for gesture control in games that use gesture recognition. They provide a BiLSTM for motion recognition in virtual reality that is fully equipped with a convolutional network (FCN). The authors suggest utilizing Leap

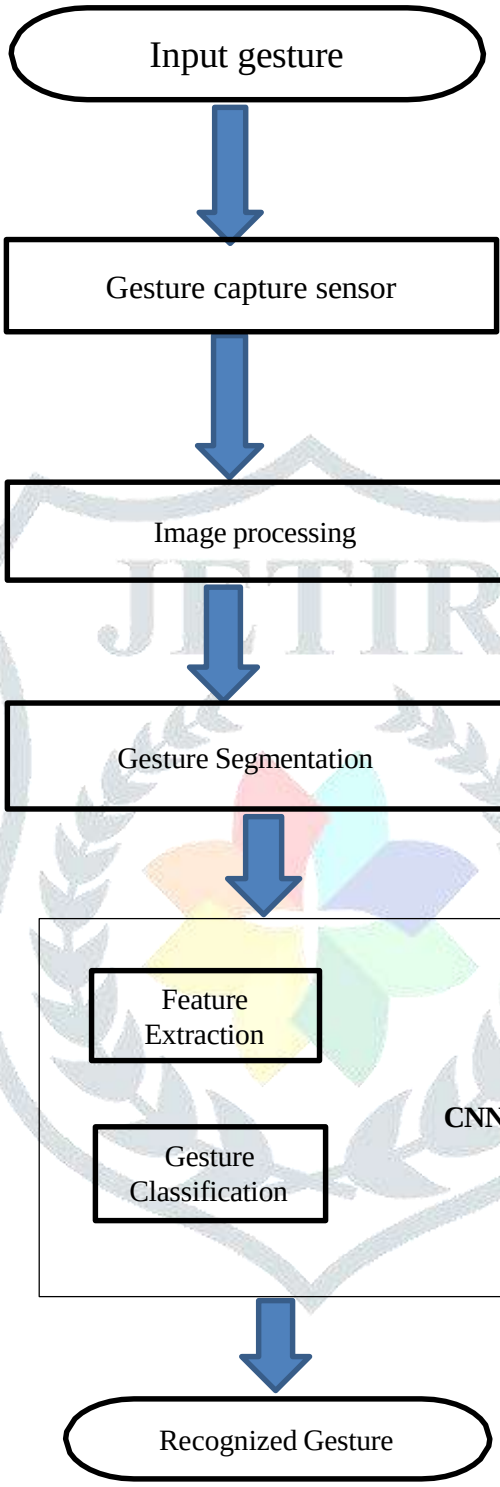
Motion Controller data to create a hand gesture recognition system. Consequently, feature extraction is done by hand. Five normalized measurements between each fingertip and the palm's center made up the characteristics. They give an example of how to use gesture recognition in games to control gestures by leveraging CNN and camera data. They offer a fully convolutional network (FCN)- equipped BiLSTM for motion recognition in virtual reality. The hybrid model developed by the authors uses the Leap Motion Controller to recognize hand gestures. This model uses an LSTM classifier and a BiSTM classifier, in that order. Lastly, they suggest the hybrid HBU-LSTM version. RIT and Leap Gesture DB, two publicly accessible datasets, are used to test the model. The Leap Gesture DB dataset contains eleven different gestures, including thumb up, index left or right, grip in or out, thumb up, index left or right, and hand swipe left or right. Twelve distinct gestures are displayed in the RIT dataset, which includes the following: grip, release, swipe, one-finger tap, two-finger tap, wipe, check mark, and pinch. Within this framework, this study suggestsutilizing data taken from the Leap Motion Controller to evaluate both human and automatic feature extraction methods forthe hand gesture identification challenge. Statistical features including wavelength, detector log (LD), pulse percentage rate (MYOP), and BiLSTM are used in the manual extraction process.

2.1 Dataset Building:

A Leap motion controller was the toolused to collect the data (LMC). Because it is less expensive and compact, we used thisdevice. Its purpose is to follow the hand. Two depth cameras and three LED sensorsare features of the LMC. Additionally, thisapparatus retrieves hands and fingers' spatial positions, directions, and velocities using the coordinate axis, the centre of which is the apparatus. Because of its low latency software and120Hz refresh rate, Leap Motion's softwarecan track 27 different hand elements, such as bones and joints, even when they are obscured by other hand parts. This is because the time lag between motion and photon is less than the human perception threshold1. Every user additionally generated five new gestures. These included the pinch, wave in,wave out, open hand, and fist. Five secondswas the allotted time for the user to executethe gesture. In this way, the user couldexecute the gesture- representing hand motion at any point within the five seconds.

Figure 1.1: Types of Hand Gesture

3. Methods:



3.1 Convolutional Neural Network:

Compared to other neural network types, convolutional neural networks perform better when given inputs in the form of speech, visual, or audio signals. A CNN is a neural network that consists of several convolutional layers. A convolutional layer is a small logistic regression where the convolution mask determines the weights and the input data values determine the constants. As the mask, a vector or a matrix can be used. A matrix is created when the input data are two dimensional (2D); a vector is created when the input data are one dimensional (1D). The output of the convolutional layer can have the same dimensions

as the input data if the stride is one and padding is applied. In padding, values of 1 or 0 are added outside the bounds of the input data; the quantity of values added is determined by the mask's size. The number of leaps required to complete a convolution is called a stride. ACNN also features a pooling layer. The dimensionality of the input data is decreased by this layer. In the end, a CNN yields a vector that is less than the input data and has a respectable abstraction or representation of the input data.

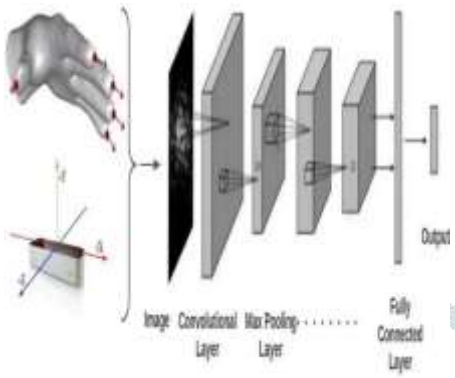


Figure 1.2 Convolutional Neural Network

3.1.1 Convolutional layer:

The majority of the calculation takes place in the convolutional layer, which is the central component of a CNN. It is necessary to have input data, a filter, and a feature map. Let's suppose for the purposes of this discussion that the input will be a color image composed of a 3D pixel matrix. This implies that an image's RGB values will line up with the input's three dimensions (height, width, and depth). In addition, we have a feature detector—also referred to as a kernel or filter—that looks for the feature in the image by scanning its receptive fields. We call this process a convolution.

3.1.2 Pooling layer:

By reducing the number of parameters in the input, down sampling, also known as pooling layers, reduces dimensionality. In contrast to the convolutional layer, the pooling operation uses a weightless filter to sweep across the entire input. Instead, the morals in the receptive field are processed by the kernel through an aggregation function, after which the output array is filled.

3.1.3 Fully-connected layer:

The name "full-connected layer" aptly describes its nature. The pixel values of the input image are not directly related to the output layer when using partially linked layers, as was previously mentioned. However, in the fully-connected layer, each output layer node is directly linked to every other node in the layer. This layer categorizes the data using the features that were extracted from the previous levels and their different filters. To properly classify inputs and produce a probability between 0 and 1, FC layers often use a softmax activation function, whereas convolutional and pooling layers typically use ReLu functions.

3.2 Bidirectional LSTM ((BiLSTM)Layer:

Convolutional neural networks (CNNs) and bidirectional long short-term memory (BiLSTM) networks are combined in a hybrid architecture for hand motion identification. Hand gesture input photos are first preprocessed for feature enhancement. The preprocessed images arethen subjected to a CNN to extract spatial features, which captures important patternsand spatial information. After that, a BiLSTM network receives the CNN's output in order to recognize the dynamic character of hand movements over time anddetect temporal relationships in the series offrames. In real-time gesture recognition tasks, the integrated model achieves excellent accuracy by classifying the motions based on the temporal context andlearning attributes.

4. Experiments:

An Alien ware PC running Windows 10 with 32 GB of RAM, six cores, and twelve 3.4 GHz Core i7 logical processors from the sixth generation was used for the experiments. The dataset outlined in the preceding section served as the foundation for the experiment design. We assessed the precision of hand gesture testing and recognition in each experiment. Furthermore, throughout the experiment's execution, the categorization test's processing time was recorded. Time is a crucial factor to consider when assessing recognition algorithms. This is as a result of the real-time applications for these methods. In the context of hand gesture recognition, real-time refers to receiving an answer from the algorithm within a millisecond or less following the execution of the gesture.

In this paper, two approaches were taken. Features could be extracted automatically or manually using one of two ways. For the manual feature extraction process, statistical functions were incorporated, while a CNN and a BiLSTM were utilized for the automatic feature extraction. The same features were analyzed by Softmax and SVM classifiers.

The following feature extraction function was used in this work for manual feature extraction: the leap motion controller signal's modification of the pulse percentage rate (MYOP); A precise indicator of applied force is the detector log (LD); Standard deviation (SD) is the square root of the average of the difference between the squared adjacent values. Wavelength (WL) can be calculated by taking the square root of the average of the difference between the squared adjacent values. An extension of WL is called enhanced wavelength (EWL). A p-value is a value that is used to select a subset of the signal. Following the use of the sequential feature selection procedure, these feature extractions were produced. Fifteen feature extraction functions were used as inputs in this feature selection technique. The scientific literature lists these 15 statistical functions of central tendency as the most often used feature extraction functions.

A signal composed of the orientations and positions of the fingers was used for the studies. The information from the five fingers was used in the recognition and classification algorithms. All these fingers also represented the data from the three channels: X, Y, and Z. Additionally, we used a five-fold k-fold for cross-validation. Window splitting was used to extract the data and feed it to the classifiers. The signal was split into 20-window windows using a step size of 15. After receiving each data window, the classifier generated a label. Eventually, a vector of labels was produced. By using a majority vote, it returned the label that was repeated the most times; nevertheless, in order for each window to be identified, it was recorded and connected to the exact instant the motion started. In this way, the signal was recognized as both a gesture and a place to rest.

The paper will also present the average time after the gesture was conducted by the algorithm before returning a response. In the test dataset, this time was recorded for every sample. When the sample was submitted to the algorithm for the classification test, the clock began, and it ended when the classifier produced a result. In addition, the data were randomly mixed before being sent into the algorithms. CNN, BilSTM, and SVM were the algorithms employed in this instance.

Table.1 Manual Feature Extraction

Algorithm	Classification Testing	Average Time of Classification testing	Recognition
CNN SVM	93.936 93.370	56 ms	84.227 81.863

We performed hand gesture detection for automatic feature extraction with the CNN. The network was fed a time series with 75 observations and 30 features. The thirty features were composed of fifteen features that represented the X, Y, and Z spatial coordinates of each finger and fifteen directions that represented the points of each finger. This indicates that the CNN was given a 1D architecture using the original 30 × 75 tensor.

A stochastic gradient descent with momentum (sgdm) method was applied in order to maximize the cost function. The CNN's parameters are modifiable. The average accuracy of the network was increased by modifying these parameters. The CNN architecture in this paper was built using MATLAB.We establish parameters like piecewise learning rate schedule. We were able to lower the training learning rate thanks to this setting. The learn rate drop factor = 0.2 was linked to this parameter. Because of this multiplicative parameter, we were able to reduce the learning rate after a predefined number of epochs.

Table.2 Automatic Feature Extraction

Algorithms	Classification Testing	Average Processing Time	Recognition
CNN-softmax CNN-SVM	99.971 99.911	30 ms	89.015 91.153

Furthermore, the model did not include the fully connected layer, normalization layer, softmax, or classification layer because the main goal of the work was to extract the features that the convolution blocks abstracted. In this way, we broke into the CNN architecture after it extracted the features and abstracted the issue, then sent them to the fully connected layer.

The input sequence layer, seven convolution layers, seven normalization levels, seven ReLU layers, five

pooling layers, two dropout layers, one flatten layer, one fully connected layer, one softmax layer, and a classification layer are the 34 layers that comprise the Bilstm. The layers bearing the convol symbol showed either a 1 × 3 convolution with 8, 16, 32, and 64 filters, respectively, or a vector of weights. Furthermore, the normalization factor for each normalization layer was set at 0.1. In a similar manner, the pooling layers used a pool of 5 and jumps of 1 while using the max function. Additionally, a regularization factor was included in the dropout layer's configuration to prevent overfitting. Ultimately, 128 gates were displayed by the BiLSTM layer.

.

Table.3 Automatic Feature Extraction

Algorithm	Classification Testing	Recognition
BilSTM-softmax	96.661	92.331

Additionally, the characteristics produced automatically by BiLSTM were acquired. These characteristics were acquired at the BiLSTM layer's output. These characteristics fed a classifier that used CNN. An SVM-based classifier was fed in the same manner.

Table.4 Automatic Feature Extraction

Algorithm	Training	Classification Testing	Average Processing Time
BilSTM-CNN	99.999	99.993	25ms
BilSTM-SVM	99.999	99.891	

Since the CNN result showed the highest accuracy, it was filtered before being subjected to a pairwise analysis. Additionally, the hypothesis contrast statistic demonstrates that BiLSTM- CNN exceeded CNN and that there was a substantial difference between the two. In addition, a distinction was noted between the two approaches' automatic feature extraction and human feature extraction.

		Classification Testing	Recognition	Processing Time
CNN SVM	Manual Feature Extraction	93.936 93.370	84.227 81.863	56ms
CNN-softmax CNN- SVM	Automatic Feature Extraction	99.971 99.911	89.015 91.153	30ms
BilSTM-softmax BilSTM-CNN BilSTM-SVM	Automatic Feature Extraction	96.661 99.993 99.891	92.331	25ms

Table.5 Summary of the performance of simple and complex models.

5. Conclusion:

This research examines the recognition, processing time, and classification test accuracy of a hand gesture recognition model. The LMC obtained the data that were needed to assess the model. Two models, one with automatic feature extraction and the other with manual feature extraction, were used to measure processing time and accuracy. Statistical functions of central tendency such as MYOP, LD, WL, EWL, DASDV, and SD were used for manual feature extraction. The same authors' earlier work served as the basis for the selection of these functions. It was also crucial to keep the feature extraction functions in the same order because the combination yields the best results. Our three-layer, fully-connected neural network model yielded an overall high accuracy result of 99.993%. In the future, we plan to integrate more datasets from real-life images with a wider range of motions to expand our model's potential uses. In order to assess our model's performance, we also plan to test it against other models, such random forest. In addition to the aforementioned, we intend to use several augmentation methods. Domain adaptation is one method that can effectively generalise the model to real photos.

Regarding the amount of parameters and layers that needed to be adjusted, it was discovered that there was no appreciable difference between the two complex models that were evaluated simple with manual feature extraction and complex with automatic feature extraction. It was noted that the simple and sophisticated models differed significantly from one another. The BiLSTM-CNN model includes multiple layers and memory, while the CNN model is a deep model with few layers. It's important to comprehend this distinction.

Researchers and developers can improve accessibility, user experience, and interaction capabilities in several fields by utilizing CNNs to push the boundaries of gesture detection systems. However, further research and development are required to address concerns including dataset diversity, model robustness, and real-world deployment considerations in order to guarantee the effectiveness and reliability of CNN-based gesture analysis systems in practical applications.

References:

[1] Ruben E. Nogals and Marco E. Benalcazar, M.E Hand Gesture Recognition Using Automatic Feature Extraction and Deep Learning Algorithm with Memory, Big Data Cogn. Comput.2023, 7, 102.

[2] Rina Damdoo, Kanak Kalyani ,Jignyasa Sanghavi.Adaptive Hand Gesture Recognition System Using Machine Learning Approach, Biosc.Biotech.Res.Comm. Special Issue Vol 13 No 14 (2020) Pp-106-110.

[3] Xiangui Bu, Human Motion Gesture Recognition Algorithm in video Based on Convolutional Neural Features of Training Images, Digital Object Identifier 10.1109/ACCESS.2020.3020141.

[4] A. Mohanarathinam,K.G.Dharani,R.Sangeetha,G.Aravindh,P.Sasikala, Study on Hand Gesture Recognition by using Machine Learning, IEEE Xplore Part Number: CFP20J88-ART; ISBN: 978-1-7281-6387-1.

[5] Hung-Yuan Chung, Yao-Liang Chung, Wei-Feng Tsai, An Efficient Hand Gesture Recognition System Based on Deep CNN, 978-1-5386-6376-9/19/$31.00 ©2019 IEEE.

[6] Nan Ma, Zhixuan Wu, Yiu-ming Cheung, Yuchen Guo, Yue Gao, Jiahong Li, and Beijyan Jiang, A Survey of Human Action Recognition and Posture Prediction, DOI: 1 0. 2 6 5 9 9/T
ST. 2 0 2 1. 9 0 1 0 0 6 8 Volume 27, Number 6, December 2022.

[7] D. K. Ghosh and S. Ari, "Static Hand Gesture Recognition Using Mixture of Features and SVM Classifier," 2015 Fifth International Conference on Communication Systems and Network Technologies, Gwalior, 2015, pp. 1094-1099, doi:10.1109/CSNT.2015.18.

[8] T. Mantecón, C.R. del Blanco, F. Jaureguizar, N. García, “Hand Gesture Recognition using Infrared Imagery Provided by Leap Motion Controller”, Int. Conf. on Advanced Concepts for Intelligent Vision Systems, ACIVS 2016, Lecce, Italy, pp. 47-57, 24-27 Oct. 2016. (doi: 10.1007/978-3-319-48680-2_5).

[9] Normani, N.; Urru, A.; Abraham, L.; Walsh, M.; Tedesco, S.; Cenedese, A.; Susto, G.A.; O’Flynn, B. “A Machine Learning Approach for Gesture Recognition with a Lensless Smart Sensor System. In Proceedings of the 2018 IEEE 15th International Conference on Wearable and Implantable Body Sensor Networks (BSN), Las Vegas, NV, USA, 4–7 March 2018; pp. 4– 7.

[10] Ameur, S.; Khalifa, A.B.; Bouhlel, M.S. A novel hybrid bidirectional unidirectional LSTM network for dynamic hand gesture recognition with Leap Motion. Entertain. Comput. 2020, 35, 100373.

[11] Núñez, J.C.; Cabido, R.; Pantrigo, J.J.; Montemayor, A.S.; Vélez, J.F. Convolutional Neural Networks and Long Short-Term Memory for skeleton-based human activity and hand gesture recognition. Pattern Recognit. 2018, 76, 80–94.

[12] Sahoo, J.P.; Ari, S.; Patra, S.K. Hand gesture recognition using PCA based deep CNN reduced features and SVM classifier. In Proceedings of the 2019 IEEE International Symposium on Smart Electronic Systems (iSES) (Formerly iNiS), Rourkela, India,16–18 December 2019;
pp. 221–224.

[13] Ikram, A.; Liu, Y. Skeleton based dynamic hand gesture recognition using LSTM and CNN. In Proceedings of the 2020 2ndInternational Conference on Image Processing and Machine Vision, Bangkok, Thailand, 5–7 August 2020;

[14] S.Gowtham, G.Dharani, DR.G.Singaravel, ―Pharmaceutical Guidance System Based on Sentimental Analysis of Chemist Reviews Using Machine Learning‖ in International Journal of Innovative Research in Computer and Communication Engineering, Volume no:10, Issue:12, Pages: 8664-8668