Hand Gesture Recognition

Helder Dias
MECD
IST, Lisbon, Portugal
IST146541

Netra Kulkarni
MECD
IST, Lisbon, Portugal
IST1101226

Pedro Zenário
MECD
IST, Lisbon, Portugal
IST102348

Abstract—**Hand gestures are an important part of our communication. We, consciously or sub consciously, use them to express ourselves better and hence interact efficiently. This makes them suitable for human-computer interfaces too. Our project focuses on identification and classification of 15 such gestures read through an ordinary camera (sensor) employing the concept of transfer learning on a pre-trained multi-channel deep convolution neural network (MC-DCNN). The sensed image frames are transformed into a 3D hand-skeleton data set using Google's open-source framework "Mediapipe" before feeding to the model for training and prediction. Further usage of these hand gestures as commands is out of scope for this project.**

I. Introduction

Hand gesture recognition is a sub-discipline of "computer vision" that concerns the interpretation of human gestures via mathematical algorithms. Gestures, the most commonly used non-verbal mode of communication for humans, can originate from any bodily motion or state - however the ones from face or hands are essentially most prominent of all. These gestures are captured through sensors and put to use as human-generated-computer-commands using machine learning and deep learning models [1].

This can be seen as a distinctive way for computers to begin understanding human body language, thus building a richer bridge between humans and machines than the primitive methods including keyboard or mouse based Text-UI or GUI.

Commonly used sensors for Hand gesture recognition include:

- Wearable Sensors (Gloves)
- Computer vision

Wearable sensors comprise of different sensor types such as flex, tactile, accelerometer and gyroscope that sense variety of physical properties. They are relatively expensive, which makes its simpler alternatives more attractive.

Computer Vision technology makes the use of cameras RGB, Depth, TOF (time of flight), Infrared, Thermal and/or Stereo to capture various kinds of images which then can be processed to extract features such as skin color, skeleton, depth, 3D model, motion etc. for further use.

Hand gesture recognition technology, overall, is proving to be a breakthrough in multiple areas such as robotics, gaming, surgery, security systems and many more.

A lot of research is already underway to better the features extraction from sensed images (for ex. with scene background limitations, unsuitable illumination conditions) and prediction algorithms simplifying the implementation further.

This project is based on the works mentioned on one of such research papers titled "Deep Learning for Hand Gesture Recognition on Skeletal Data" [2].

II. Problem Description

"We need a model to identify or essentially classify a human hand-gesture sensed through an ordinary camera into one of the 15 predefined categories with a fair level of accuracy and processing time"

Recognition of human gestures comes within the more general framework of pattern recognition. This can be subdivided into two processes:

- the acquisition process which converts the physical gesture to numerical data, and
- the interpretation process, which adds meaning to it.

A. Why is the problem important?

Hand gestures can **add another dimension to effective communication** between humans, and also in the context of human computer interaction.

While **hygiene** remains one of the most significant measures in the current time, gesture intent interpretation is a key to human-machine-interaction especially in the situations that **do not endorse touch**.

Hand gesture recognition can significantly aid **sign language** users with impaired hearing or speech to communicate.

Hand gestures have been of great interest in the surgery rooms as it helps masked **surgeons communicate while keeping their hands sterile**. Robotics in medical science is picking up and **doctors interaction with these assistant robots during a surgery** can be significantly befitted using Hand gesture recognition systems.

III. HOW DID WE ADDRESS THE PROBLEM?

A. What did we start with?

1) ***A Pre-trained Deep Model trained on 14 hand gestures****:* The paper proposes use of multichannel convolutional network with two feature extraction modules and a residual branch per channel as shown in the figure 1.

The data will be fed in multiple, fixed-length, 1D sequences (s_1, s_2, ..., s_c). Each channel has 3 branches - two of which have similar architecture designed for feature extraction. The input is passed to a convolution layer, whose output is subsampled using a pooling layer. This process is repeated two more times.

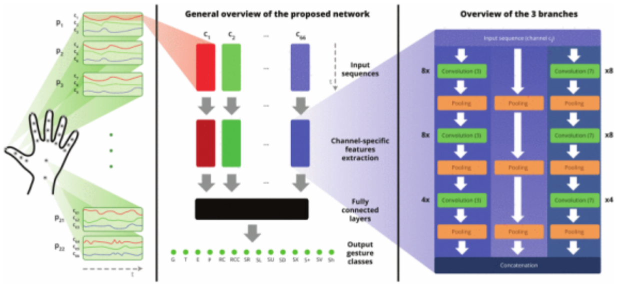

General overview of the proposed network
Overview of the 3 branches
Input sequences
Channel-specific features extraction
Fully connected layers
Output gesture classes
Concatenation

Fig. 1: *Proposed parallel convolutional neural network.*

For a single branch, the difference between all the three convolutions resides in the number of feature maps used; the difference between the two branches resides in the size of the convolutions kernels. Having two kernel sizes for the time convolution layers allows the network to directly work at different time resolutions.

The third branch in a channel works as a residual branch. Residual branches make it easier to optimize networks using a better gradient back-propagation in the training phase; also empirically improving the accuracy.

*2) **The 3D hand-skeleton dataset for 14 hand-gestures that the model was trained on**:* The Dynamic Hand Gesture-14/28 (DHG) dataset was created and introduced, [3] as a part of the SHREC2017-3D Shape retrieval Contest, as suggested in the paper. It consists of 3D hand skeletal representations returned by the Intel RealSense depth camera, corresponding to the 3D coordinates of 22 landmarks of the human hand, see figure 2.

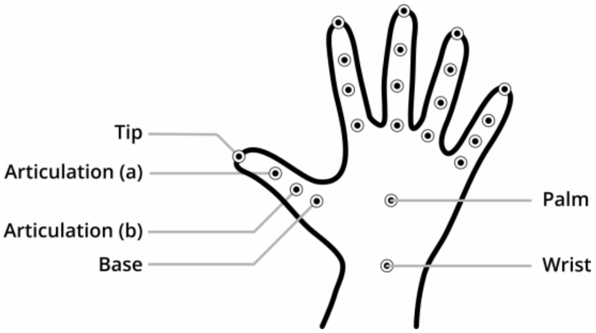


Fig. 2: *Hand skeleton returned by the Intel RealSense camera. Each dot represents one of the n=22 joints of the skeleton.*

*3) **Mediapipe framework**:* MediaPipe Hands is a high-fidelity hand and finger tracking solution [4], [5]. It employs machine learning (ML) to infer 21 3D landmarks of a hand from just a single frame, as shown in figure 3.

B. Our Adaptation

With a multi-channel deep convolutional neural networks (MC-DCNN) model at hand and a ready-to-use hand skeleton dataset, our obvious approach comprised of retraining the model with an extra hand-gesture, as per our plan. While re-training from scratch was an option, Transfer learning approach appeared faster and more suitable for adding 1 new gesture - a small data set.

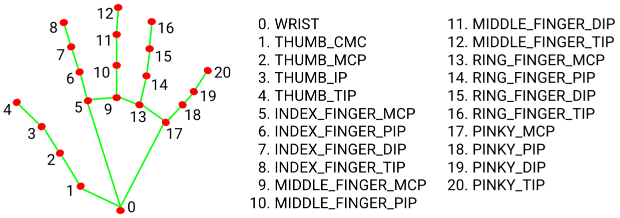


Fig. 3: *Hand skeleton returned by the Mediapipe. Each dot represents one of the n=21 joints of the skeleton.*

TABLE I: *List of gestures*

#	Name of the gesture
1	Grab
2	Tap
3	Expand
4	Pinch
5	Rotation Clockwise
6	Rotation Counter Clockwise
7	Swipe Right
8	Swipe Left
9	Swipe Up
10	Swipe Down
11	Swipe X
12	Swipe +
13	Swipe V
14	Shake
15	New Gesture = ”Rock on” Sign

Table I lists the 15 gesture classes.

The hand gesture we chose is the famous ”Rock on” salute - Index finger up, middle fingers down, pinky up, thumb in or out.

The steps we followed can be summed up as:

- **Data Capture** - a new hand gesture using our laptop camera.
- **Data Pre-process** - transform the captured image frames into 3D hand skeleton points dataset.
- **Preparing the pre-trained model for transfer learning** - adjust the last fully connected layer of the model and freeze the others
- **Train the model** - using the hand gesture data for 15 gestures
- **Validate** - hyper-parameter tuning and validating the accuracy
- **Test the model** - predict using test data and evaluate the final accuracy.

1) Data Capture: We captured 100 frames of our new hand gesture ”The Rock on sign” with an ordinary laptop camera using the ”mediapipe” framework [6] such that the hand image is transcribed into a 3D hand skeleton of 21 data points, as in figure 4. Manual augmentations were performed such as thumb in, thumb out, index finger and pinky more open or restricted and Little rotation on the hand - to collect more samples for our training and test datasets.

2) Data Pre-process: The dataset that our model was trained on had 22 datapoints as opposed to 21 on the our capture. We employed centroid calculation 1 for the missing ”Palm” landmark (using 3 other landmarks, wrist, base of

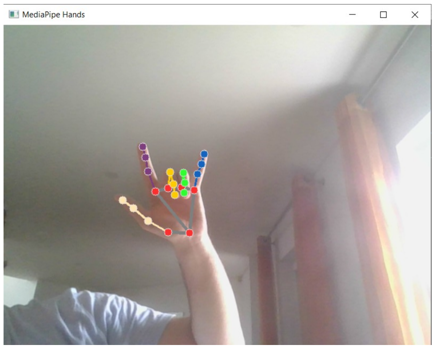


Fig. 4: *Our new "rock on" gesture with skeleton outlined by mediapipe.*

index finger and base of little finger) to get the data prepared for training.

The centroid of a finite set of k points $x_1, x_2, \ldots, x_k$ in R^n is

$$\mathbf{C} = \frac{\mathbf{x}_1 + \mathbf{x}_2 + \cdots + \mathbf{x}_k}{k} \tag{1}$$

Every point has 3D coordinates such that each training record is now of size 100*66 (=22 * 3 coordinates). The dataset was then pickled into train and test subsets.

3) Model Preparation: The pre-trained model was loaded for transfer learning. The first 66 hidden layers were frozen while the last fully connected layer was modified to now accommodate 15 gestures instead of 14.

4) Train the model: The model was then trained using transfer learning over multiple iterations and compositions of the 15-gesture dataset. The different approaches and outcomes are discussed in the results section.

5) Validate: Multiple iterations of tuning the model helped increase the accuracy as well as insights into the data compositions mentioned earlier.

6) Test the model: Finally the model was tested on a test subset as well as freshly acquired human hand gestures.

IV. Results & Discussion

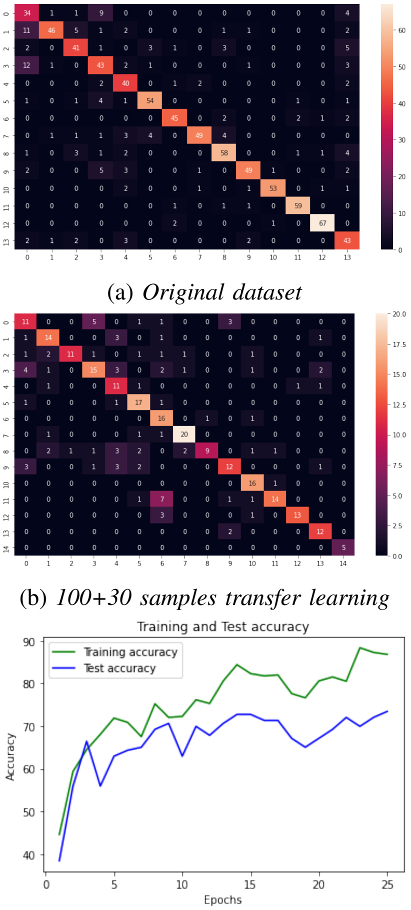
Validation, tuning and testing the model demanded multiple iterations. This section will list out results of a few important iterations - in fact milestones in this journey.

A. As-it-is Pre-trained Model: Original dataset

The confusion matrix for the crude pre-trained model on a test set can be seen in figure 5a. It is almost diagonal without significant miss-classifications but gestures 2 (Tap) and 4 (Pinch) are sometimes miss-classified as gesture 1 (Grab), whereas gesture 1 is sometimes classified as gesture 4.

B. Re-training the Model

We have primarily focused on training data size and composition for our iterations. Usual methods of hyperparameter tuning used like varying learning rates, batch sizes and early stopping were pondered up on too.



(a) *Original dataset*

(b) *100+30 samples transfer learning*

(c) *100+30 samples model accuracy*

Fig. 5: *Confusion Matrices and model accuracy*

While 66 feature extraction layers, in our case, were frozen so that only the last fully connected layers of the model could be trained, lowering the learning rate (that usually helps while training hidden layers) did not seem to be of a great use. Our accuracy levels with a fixed learning rate were looking sub par; the number of weights to train were high; and we used a small dataset to train with - unchanged learning rate helped us sail through the training faster.

While the 6 feature extraction layers (DCNN), in our case, were frozen so that only the last fully connected layers of the model could be trained, lowering the learning rate did not seem to be of a great use. Our accuracy levels with a fixed learning rate were looking sub par; the number of weights to train were high (7128*1960*15); and we used a small dataset to train with - unchanged learning rate helped us sail through the training faster.

Transfer learning usually needs a smaller dataset so we decided that a good starting point was to select only ≈10% of the available data for each class plus the same amount of samples from the new gesture.

1) Small and balanced data set: Proceed to retrain the model with small set of 20 samples of each class. In this setup, the training accuracy increased until 90%, but the test accuracy stayed very low (<50%). With these results, our model was over-fitting to the training set but not performing good with general data.

2) Medium but imbalanced data set: Overfitting can be remedied by either applying regularization techniques or by increasing the training data. We decided to go with the latter and increased the samples to 100 for each of the old gestures and 20 for the new gesture. This was based on the observation that the model was performing well on the new gesture and suffered with the old gestures. The test accuracy increased this time to more or less 65%, but we thought that we could do better.

3) Medium and more balanced dataset: So, we captured 10 more samples of the new gesture to try and balance the dataset, and turns out we did the right thing, since the accuracy test increased once again, this time to around 75%, as seen in figure 5c. The confusion matrix of the new model, in figure 5b is almost a diagonal with only a few miss-classifications, the most common being the miss-classification of 12 (Swipe +) with 7 (Swipe Right), and more dispersed than what was seen in the original model.

V. DECISION HIGHLIGHTS

Some of the key decision-making during this project involved a lot of desk research and brainstorming. Here we list a few of them that defined the course of the path we took:

A. Use Transfer Learning or Train from scratch?

The paper and the model we handpicked for our project gave us access to the crude model as well the original dataset. This tempted us to train the model (tweaked on the last layer) from scratch for 15 gestures now. Should we choose to go this direction, feature selection on the 14 data gestures data - especially removing the missing ”Palm” landmark in the mediapipe that we redundantly calculated using some of the other feature points - would benefit the model complexity. However Transfer learning was a favoured choice specially because it works with small data set, is faster and easier to re-train.

B. Feature Selection or Pre-Processing for the missing Palm landmark?

Feature selection helps reduce model complexity especially through shorter training time and smaller space-usage. While it is attractive, it is not feasible with transfer learning to drop a few features on the re-train dataset. Hence, it was an informative choice to preprocess data and add the redundant missing data point, as discussed earlier.

C. Data Augmentation using SMOTE Techniques?

Getting 30 training records of the same hand gesture manually is a cumbersome task and hence, our natural quest to find suitable techniques for data augmentation commenced. However, we soon realised that the way mediapipe works - where it directly sends us the hand landmarks and not the actual image info - techniques like SMOTE could not be used.

D. Does the model over-fit if re-trained on the same 14 gestures?

In order to verify this, we tried re-training our pre-trained model on a dataset with just 20 samples from each of the 14 gestures in the original train dataset and re-trained the model. The test result on a totally new data set was 80%. This trial negated our fear and indicated to continue going forward with our approach.

E. Do we need architectural changes?

The original architecture was based on the features of 22 hand landmarks, since the palm landmark can be calculated from other landmarks, being redundant, the number of input features could be reduced to 21 (63 channels). That would result in a more compact model but would require a full training approach instead of transfer learning.

The classification layer of the model, is composed of a fully connected layer with 1936 nodes (output of the convolution layers is flattened to 7128) followed by a final 14 node layer corresponding to one of each class. Other forms could be considered, such as SVM, decision trees, fewer nodes, and even removing the first layer in the classifier. The decision was to maintain the architecture because there’s a great difference between the CNN layers’ flattened output (7128) and the number of output nodes. Also, in deep learning architectures, fully connected layers are the usual choice to be used for classification, and SVM typically performs subpar [7]. The new classification layer would only need to be extended to 15 output nodes instead of 14.

VI. CONCLUSION

A pre-trained deep learning model trained on hand gesture dataset was handpicked, and transfer learning was applied to train it to additionally be able to classify a new gesture making it 15 gestures in total. Multiple iterations of the train-validate-test process were employed in order to reach acceptable model accuracy levels. Our approach mainly focused on varying data sizes and composition to get to a final accuracy level of 75%. We can successfully conclude here that training a pre-trained model with fraction of the original data-size helped us build a model with accuracy comparable to the original one, and an added capacity of classifying a new gesture.

REFERENCES

[1] M. Oudah, A. Al-Naji, and J. Chahl, “Hand gesture recognition based on computer vision: a review of techniques,” *journal of Imaging*, vol. 6, no. 8, p. 73, 2020.

[2] G. Devineau, F. Moutarde, W. Xi, and J. Yang, “Deep learning for hand gesture recognition on skeletal data,” in *2018 13th IEEE International Conference on Automatic Face & Gesture Recognition (FG 2018)*. IEEE, 2018, pp. 106–113.

[3] Q. D. Smedt, H. Wannous, J.-P. Vandeborre, J. Guerry, B. L. Saux, and D. Filliat, “3D Hand Gesture Recognition Using a Depth and Skeletal Dataset,” in *Eurographics Workshop on 3D Object Retrieval*, I. Pratikakis, F. Dupont, and M. Ovsjanikov, Eds. The Eurographics Association, 2017.

[4] F. Zhang, V. Bazarevsky, A. Vakunov, A. Tkachenka, G. Sung, C.-L. Chang, and M. Grundmann, “Mediapipe hands: On-device real-time hand tracking,” *arXiv preprint arXiv:2006.10214*, 2020.

[5] Google, “Mediapipe hands,” 2022, accessed: 2022-01-06. [Online]. Available: https://google.github.io/mediapipe/solutions/hands.html

[6] ——, “Mediapipe in python,” 2020, accessed: 2022-01-06. [Online]. Available: https://google.github.io/mediapipe/getting-started/python.html

[7] A. F. Agarap, “An architecture combining convolutional neural network (CNN) and support vector machine (SVM) for image classification,” *CoRR*, vol. abs/1712.03541, 2017. [Online]. Available: http://arxiv.org/abs/1712.03541