MirrorGen

Wearable Gesture Recognition using Synthetic Videos

by

Arun Srivatsa Ramesh

A Thesis Presented in Partial Fulfillment
of the Requirements for the Degree
Master of Science

Approved July 2018 by the
Graduate Supervisory Committee:

Sandeep Gupta, Chair
Ayan Banerjee
Yezhou Yang

ARIZONA STATE UNIVERSITY

December 2018

ABSTRACT

In recent years, deep learning systems have outperformed traditional machine learning systems in most domains. There has been a lot of research recently in the field of hand gesture recognition using wearable sensors due to the numerous advantages these systems have over vision-based ones. However, due to the lack of extensive datasets and the nature of the Inertial Measurement Unit (IMU) data, there are difficulties in applying deep learning techniques to them. Although many machine learning models have good accuracy, most of them assume that training data is available for every user while other works that do not require user data have lower accuracies. MirrorGen is a technique which uses wearable sensor data and generates synthetic videos using hand movements and it mitigates the traditional challenges of vision based recognition such as occlusion, lighting restrictions, lack of viewpoint variations, and environmental noise. In addition, MirrorGen allows for user-independent recognition involving minimal human effort during data collection. It also helps leverage the advances in vision-based recognition by using various techniques like optical flow extraction, 3D convolution. Projecting the orientation (IMU) information to a video helps in gaining position information of the hands. To validate these claims, we perform entropy analysis on various configurations such as raw data, stick model, hand model and real video. Human hand model is found to have an optimal entropy that helps in achieving user independent recognition. It also serves as a pervasive option as opposed to a video-based recognition. The average user independent recognition accuracy of 99.03% was achieved for a sign language dataset with 59 different users, 20 different signs with 20 repetitions each for a total of 23k training instances. Moreover, synthetic videos can be used to augment real videos to improve recognition accuracy.

To my family, roommates, friends and colleagues.

ACKNOWLEDGMENTS

I would like to express my sincere gratitude towards my advisor Dr. Sandeep Gupta in supporting my MS research. I would like to thank Dr. Ayan Banerjee for his constant guidance, support, and knowledge. I would like to thank Dr. Yezhou Yang for his support for my research. Their active feedback and responses helped me complete the dissertation procedures successfully.

I would like to thank Ph.D. students Prajwal Paudyal and Junghyo Lee for their expert knowledge, guidance, fascinating discussions and for reviewing, editing and supporting my work. I would also like to thank all the members of the IMPACT lab for their support.

This project was partly funded by NIH NIBIB (Grant EB019202) and NSF IIS (Grant 1116385). I thank Arizona Commission for the Deaf and Hard of Hearing (ACDHH) and Paul Quinn, Director of ASL at ASU for their invaluable ASL domain knowledge and feedback.

TABLE OF CONTENTS

CHAPTER Page

LIST OF TABLES

LIST OF FIGURES

Figure Page

Chapter 1

INTRODUCTION

With the advancements in deep learning techniques, the task of activity recognition is gaining more attention in the recent years. Various large scale datasets are available for the purpose of activity recognition [14, 29, 31]. However, video based approaches have some basic problems. These include problems like occlusion, where the subject or a part of the subject performing the activity is hidden from camera viewpoint and is not visible; the lack of different viewpoints for the same activity, which would greatly help any deep learning model to generalize effectively; the presence of background noise affects the confidence score of classification; Object localization issue, for example, presence of a bed in a given scene increases the confidence of the class "subject is sleeping" even though this may not be true. Humans recognize actions even if objects involved in performing the action are not present. Using a camera or an infrared (IR) camera like Microsoft Kinect might seem to be intrusive to some users and raises privacy concerns. It has a fixed setup and is also not pervasive. Another major problem is building a good quality dataset with annotations needs a significant amount of human effort which involves tasks like background removal, de-identification.

Sign Language Recognition (SLR) is a very important sub-field of activity recognition due to its impact on accessibility and gesture based Human Computer Interaction. Researchers have approached SLR from the perspective of either video based systems or wearable sensor based ones. Video based systems utilize RGB and/or depth sensors while most wearable sensors for this purpose use IMU (Inertial Measurement Units). IMU sensors use a combination of accelerometers and gyroscope, to

give raw specific force and angular rates of the mounted body.

Most of the state-of-the-art video based recognition systems with high performance use Convolutional Neural Networks (CNN) [14, 16, 17].

One of the reasons for this high performance is that when training examples are scarce such as in the case of special-case applications like SLR, these systems use models pre-trained on a more extensive dataset like ImageNet [16] and then fine-tune them for a special case application. This idea, known as transfer learning, improves performance by transferring the lower to mid-level features from one problem domain to another [21].

With recent advancements in wearable technologies, a lot of research has been dedicated to solving the problems of SLR using armband sensors such as the Myo [22, 23, 27, 38]. One of the key advantages of this approach is that wearable sensors enhance usability [1] and are resistant to classic problems associated with image/video recognition such as occlusion, lighting restrictions and environmental noise. Further, data collection for wearables is less cumbersome and more privacy preserving than for videos. Modern deep learning techniques such as CNNs are not generally used with wearable systems due to the lack of extensive training datasets and the nature of IMU data. Thus, research that focus on wearable sensor based techniques have not been able to leverage the advances in deep learning techniques effectively. Furthermore, due to the split in approach for solving the same underlying problem of gesture recognition, the datasets created have also been bifurcated. In an attempt to bridge this gap while achieving state-of-the-art accuracy, MirrorGen, a technique to convert armband orientation data into animated videos is proposed as shown in Fig. 1.1. Experimental results show that user-independent recognition accuracy for MirrorGen based system was significantly higher than other state-of-the-art user independent systems.

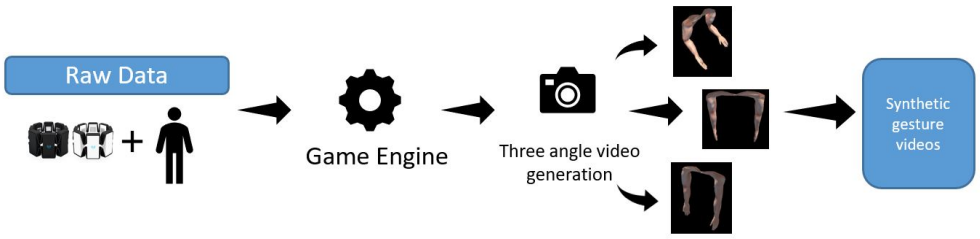

Figure 1.1: Conversion of Raw Sensor IMU Data to 3D Human Hand Model Video Frames.

MirrorGen also provides an efficient way of generating synthetic data to make the recognition algorithm robust to change of camera angles, as animations for different camera angles can be easily generated and added to the training set as explained in Section 4.1. Rather than describing gestures as time series data, the gestures are visualized as continuous frames of hand models as seen in Fig. 1.1 and inference is done by the Two Stream networks as seen in Fig. 5.2.

Gestures can be classified into two types. (a) Structured - Gestures like American Sign Language which need a structured learning system and are not easily understood by other humans who are not familiar with the gestures. (b) Unstructured/Pantomimes - Gestures which are common to all humans irrespective of cultural diversity like eating, opening a door, opening a bottle, wave, etc., [10].

The name MirrorGen was inspired from the concept of Mirror Neurons from [10] and also because the video generation technique is similar to a basic mirroring of human gesture. Mirror neurons are neurons in the human brain which activate when a person performs or when the person observes another person performing the same action. The neuron mirrors the behaviour of the other.

User independent recognition of gestures is a well known issue in the field of activity recognition. Since sensor data from all users are visually reconstructed using

the same model, all users have the same video representation of their gestures which allows effective generalization.

There is also not much video data available for specific domains like ASL. To experiment with this theory if synthetically generated data can be used to augment the limited real videos for large scale video classification, real and synthetic videos are mixed to test the transfer of mid level feature representations.

Contributions: The main contributions of this work are as follows:

1. MirrorGen: A technique to create animated videos using only wrist-worn orientation sensors that achieves state-of-the-art user independent recognition accuracy while increasing ease of use during training and testing.
2. Results on real videos using augmented synthetic data to get a mixed dataset of both real and synthetic videos for training.

This thesis report is structured as follows: First, in Chapter 2, the works related to all the domains involved are discussed. Then, in Chapter 3 the process of acquisition of data is explained. Then, in Chapter 4 the process about how MirrorGen converts orientation data from armband sensors and generates synthetic videos from them and an analysis of the generated videos is explained. Then, in Chapter 5 the different models and the system architecture for training and testing are explained. In Chapter 6, in the results section, a comparative analysis of MirrorGen technique against other machine learning techniques is performed. In the discussion section, the possible shortcomings of this technique, as well as the preliminary results for live video recognition using models trained only on synthetic videos and tested on real life videos are discussed. Finally, some of the possible future work is discussed.

Chapter 2

RELATED WORK

In this chapter, work related to wearable sensors, dataset generation, transfer learning, activity recognition and deep learning architectures suitable for activity recognition are discussed.

Wearable Sensors: Thomaz et al. [32] have used an IMU based activity recognition using smartwatches for eating activities while Chung et al. [6] have used glass-like wearables for chewing detection using temporalis muscles and Paudyal et al. [22, 23] have used armbands for sign language recognition. All of the above techniques use handcrafted features which requires domain expertise and generally do not scale well. Using deep learning to learn representations is possible, for example, Fang et al. [11] used a Leap Motion sensor to track the skeleton joints of the palm to identify ASL signs. However, this method has a restricted field of vision in front of the chest and ASL words which involve gestures like "father" that are performed near the face cannot be recognized. One of the main drawbacks of using deep learning techniques is the lack of large datasets to train on. However, this work focuses on considering a small raw sensor dataset for the task of hand gesture recognition, converting it into visual data by generating synthetic videos and using various mid level representations of videos [21] to fine tune a larger network resulting in good recognition accuracy of gestures.

Vision-based approaches: There are various vision based deep learning approaches that address the problem of SLR [4, 7]. These approaches suffer from conventional problems for computer vision like occlusion, lighting, viewpoint variation, background noise as well as privacy concerns. Using only depth sensors mitigates

some of the privacy concerns, however, there is a trade-off with accuracy and this approach is not robust to viewpoint variations [9].

Synthetic Videos: Synthetically generated data has been used to train complex CNN architectures to perform various tasks on the real world such as pose identification, learning from 3D games tracking and action recognition [12]. The PHAV dataset [8], one of the largest synthetically generated datasets is built by procedurally generating human activities and using various external factors like weather, outside lighting, and environments. The MOCAP extracts descriptors like trajectories from videos to generate videos by using a reduced number of randomly selected features. However, there are a different set of requirements for using synthetic videos for sign language recognition such as the need for viewpoint variance, visibility of the entire signing area, and proper handling of occlusion and other environmental variations.

Although these problems also exist for activity recognition, there are additional heuristics obtained from the objects present in the scene that aid in recognition. [24, 28, 36]. Due to this reason, this work uses only the trajectories of the lower arms by using the quaternions obtained from armband sensors on each arm so that clean video can be generated with only hand motion trajectories from various viewpoints. Similar to the works [2, 8, 12], a predefined 3D hand model which is a part of a 3D human model available in the Unity Asset Store is used. It includes arm joints, which facilitates the use of wearable sensors as a controller for the hand movements of the model. The state-of-the-art deep learning architectures for activity recognition [30, 35] are used to perform recognition on the generated synthetic videos.

Some works involve training the models using visual abstractions like clipart, sketches [5, 39] and the concept of Zero Shot Learning which is an extreme case of transfer learning where real world data is classified based on synthetically learned

features [25]. This work, however, has specific constraints as discussed in Section 4.2 which makes Zero Shot recognition to perform poorly.

In this case of generating videos using a game engine, hand gestures can have multiple viewpoint perspectives as opposed to generating video datasets of gestures which require all possible camera view points. This is a special case of data augmentation where the data is not augmented using random skews, flipping images, modality or color channel modification. This is an additional human knowledge infusion used to augment the training dataset for improved performance. The work on using depth information to learn side representations of the RGB image has a similar idea [34]. However, it relies heavily on the need for depth information and to hallucinate side images [13] or getting depth information by surface normal estimation technique [37].

Chapter 3

DATA ACQUISITION

In this chapter, the process of acquisition of data is explained in detail. The process of data collection, the setup of the system, user interface and the systematic variations made in the dataset are discussed.

The dataset consisted of 20 words from the American Sign Language words. Each subject is made to perform each word 20 times from 59 different users. (IRB : STUDY00004155)

These words were picked in order to introduce significant variations in hand trajectories and also included a significant amount of highly correlated signs (eg., if and father) to make sure that the model not only performs well on signs with highly different hand trajectories but also picks up on the minor trajectory variations as well. This included a mixture of 10 one and 10 two handed signs as shown in Tab. 3.1.

Around 24000 videos of Synthetic-ASL (S-ASL) are generated, which is approximately 20 hours of video and 1000 videos per category. The generated videos also include three angles of viewpoint variation left, center and right.

3.1 System Setup

The setup consists of two armbands (Myo) worn on each hand of the user. A depth camera (Microsoft Kinect) is focused on the user to give a skeletal structure feedback and RGB frames to observe ground truth position coordinates of all the body joints. The Myos are calibrated by following a rest position. This work primarily focuses on the quaternion values from the armband sensors which gives the rotation of the arm about the elbow joint. The calibration step is done by making the user start at a

Table 3.1: Dataset of 20 ASL Words Generated for the Task of Hand Gesture Recognition.

ASL	**Words**
One hand	And, Cop, Father, If, Hearing
	Cat, Go out, Deaf, Find, Gold
Two hands	Good Night, Can, Cost, Day, Hurt
	Here, About, Decide, Large, Hospital

rest position. The user stands in front of the Kinect with calibrated Myo armbands and performs the ASL words. Each word is given a fixed time of three seconds. The user starts at rest, performs the sign, goes back to rest position. Multiple users are allowed to stand in front of Kinect, move around (within the range of the Kinect) to include spatial variations.

Only the roll, pitch and yaw values from the right and left hand are used to capture the relative rotation of the arm. These rotation angles are one of the three Inertial Measurement Units (IMU) orientation data provided by the Myo armbands. The complete dataset includes all joints position of the body from the Kinect and the orientation (3 sensors), accelerometer(3 sensors), gyroscope values (3 sensors) (IMU) and electromyogram (EMG) sensors(8 sensors) from the armband sensors for each hand. The data is collected at 15 frames per second.

Summary:

Myo armbands and a depth sensor (Microsoft Kinect) are the primary sensors used to collect the data. This is an extensive dataset consisting of around 24,000 instances and will be available on the Impact Lab server for future work.

Chapter 4

APPROACH

In this chapter, the method of generation of animated videos from the raw sensor data obtained from the armband is discussed in detail. The analysis of how much error this system has when compared to the ground truth values when transforming the sensor values into videos is performed and the approaches that were used to train this system are discussed.

4.1 Synthetic Video Generation

The following subsections explain in detail the individual components involved in the synthetic video generation process.

4.1.1 Scene

The scene contains an empty synthetic environment which appears as a black background in the generated videos. A predefined set of 3D hands from a 3D human model available in the Unity Asset Store is used. Only the upper and lower arms from the 3D human model are used to generate the videos. These hand models will be performing the hand gestures based on the orientation data. The lower arm is attached to the upper arm using the elbow joint. Since only lower arm movements are considered, the rotation is on both the lower arms with respect to the elbow joint. To track the movement an in-scene camera placed at a fixed distance from the hand models is used. The camera is placed so that it gives an appearance of a third person camera.

4.1.2 Video Generation:

The 3D human model used here consists of only the left and right hands since only hand tracking and hand gesture recognition is the primary focus. The motion of the hand model is performed by rotating the hand model around the elbow joint using the raw orientation data. The raw orientation time series data obtained from the armbands are used to generate the video frames (320×240) at 30 frames per second as shown in Fig. 1.1. The hand coordinates of the 3D hand model are tracked to obtain the position data. The position data from Kinect is used to do error estimation as discussed in 4.2 and shown in Fig. 5.2.

4.1.3 Viewpoint variations

In order to introduce diversity and to account for viewpoint variations in real time datasets, the angles at which the camera points to the hand models are varied. By doing this, generation of videos of hand gestures as if the third person is viewing them from an angle is made possible. In real-world scenarios, a dataset collection process typically involves a subject standing in front of an RGB camera or a Depth sensor and performing the gestures. Either multiple cameras are placed in different angles to capture various viewpoint versions of each gesture or the person has to perform the sign multiple times at different angles to the camera. This is not only time consuming but also it is not feasible to generate a single gesture for all possible real-time viewpoint scenarios.

4.2 Synthetic Video Analysis

The following are the observations when the generated synthetic videos are analyzed.

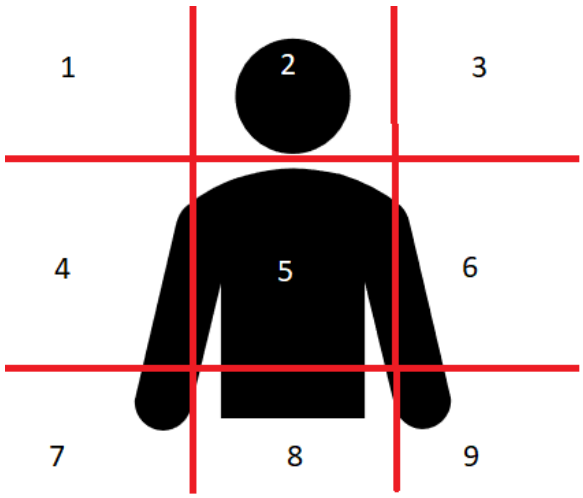

Figure 4.1: Region of Occurrences of Various Words.

4.2.1 Choice of Words:

The 20 words in this dataset are picked in such a way that they fall in various regions of interests as shown in Fig. 4.1. This subset of words serves as a sufficient representation of the words in American Sign Language since all the words involve hand trajectories which lie only inside the above region of interests. Due to the dominant right hand, region 4 has more occurrences of words as shown in Tab. 4.1. Ignoring the regions with very high and very low frequencies (caused due to dominant hand), chi-squared test to see if the frequency of signs follow a uniform distribution gives a score of 0.81.

4.2.2 Failed Generations:

While generating videos, sometimes there are noisy sensor data which leads to erroneous rotation of the lower arm. These videos were manually identified and removed. Some of the generated error cases are shown in Fig. 4.2. All the bad quality data generated were pruned in the preprocessing step.

Table 4.1: Word Occurrences of Regions Shown in Fig. 4.1.

Region	**Words**
1	And, Cat, Decide, Day, Go Out, Gold
2	And, Cat, Father, If, Hearing, Day, Deaf, Gold, Good Night
3	And, Day
4	Can, Cop, About, Father, If, Hearing, Find, Gold, Hurt, Here Hospital, Large, Decide, Day
5	About, Cop, Good Night, Cost, Hurt, Hospital, Large, Day
6	Hospital, Large, Decide, Day, Can, About, Hurt, Here
7	Can, Find, Here, Decide
8	Good Night
9	Can, Here, Decide

4.2.3 Lower Arm Side-effects:

Since only the rotation movements of the lower arm are being considered, there are ASL words with significant similarity like "father" and "if" which both have almost the same lower arm movement. So most of the signs have a significant movement of the upper arm, which are represented by the 3D hand model by moving behind the XY plane. This although ensures a unique way of representing the gestures.

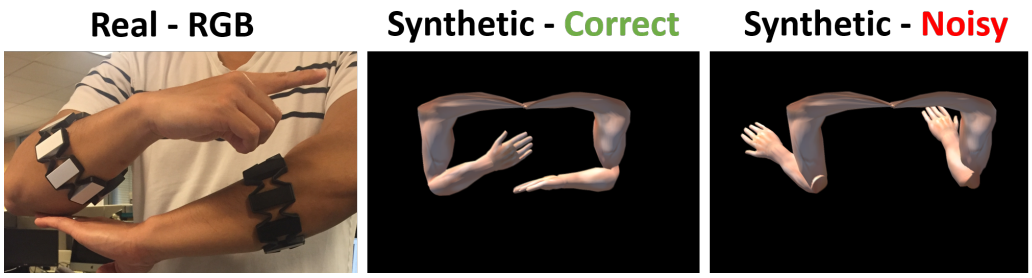

Figure 4.2: Bad Generations Due to Noise in Sensor Readings.

4.2.4 Entropy Analysis

Preprocessing with synthetic hand models helps in obtaining position data for the various locations of the hands while performing gestures. Theoretically, 3D position coordinates information can be extracted by double integrating raw accelerometer values. This technique, however, was found to introduce significant errors while obtaining displacement information caused by the gesture on the original position point. This happens because getting displacement by double integrating accelerometer values causes the errors to accumulate through the integrals. Synthetic videos help to mitigate these issues since the hand positions are calculated using orientation changes from the initial calibration point. The values can be input directly to a hand model generator such as the one in Unity Game engine to create movement and future orientation information with more fidelity and with fewer displacement errors as compared to the locations generated by using accelerometer sensors.

Real video analysis: Raw RGB videos can also be used for recognizing gestures. The recognition accuracy of 91.20% was achieved by using only raw RGB pixels is significantly lower than was achieved by using synthetic video data. There are various reasons for this: 1) Video data includes information unnecessary for gesture recognition such as background objects and their movements, and color information for

clothes and body of users, and 2) Rotation information for hands is not significantly noticeable.

A simple machine learning model for gesture recognition can also be trained with only the information from orientation time-series. However, it is found that using only orientation information does not give recognition accuracies comparable to using synthetic videos as seen in Tab. 6.1. Although the videos are created solely from the orientation sensors, synthetic videos include more information than contained in the orientation information. This is because the use of hand models provides a means of domain constrained extrapolation to create more data points. The optical flow of a synthetic video contains information not only on the point on the arm that the Myo device was worn in but it has extrapolated data points throughout the arm starting from the palm and ending at the elbow joint. This adds a significant amount of information for learning quantified as the entropy information between the optical flow of generated videos versus that of the raw sensor data as seen in Fig. 4.4. This added information is especially useful as the use of human movement model constrains the extrapolation to include only movements that are possible for a human hand. Thus, the use of human models provides an accurate and useful extrapolation which otherwise would have to be performed using mathematical equations which would be less precise and would require significant human effort.

Stick model analysis: To test the need for human hands for the synthetic video model, experiments were performed replacing the human hand with sticks as shown in Fig. 4.3. Although the generated videos using the stick models were useful, there was a loss in accuracy of 6%. Although, the generated videos using sticks instead of hands had similar entropy as seen in Fig. 4.4, the difference in recognition accuracy is explained since the stick models, due to their 2D nature, can use only the yaw, and

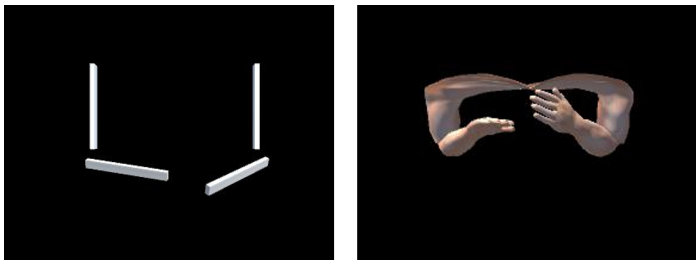

Figure 4.3: Stick Model Versus 3D Human Hand Model.

pitch information while the complete human hand model is able to capture the roll information as well. In many instances, ASL gestures are differentiated not only by the location and movement of the hands but also by the orientation as explained by [26]. Thus, the use of synthetic videos using human hand models for ASL recognition is justified.

For performing entropy analysis, the gray-scale optical flow images are flattened to represent a 1-dimensional array and Shannon Entropy is calculated for this 1D array of gray-scale values to check the distribution of information.

A single video consists of multiple frames. The 1D array is generated by averaging all the pixel values from the frames. This is done for each word video and is shown in Fig. 4.4. The accelerometer raw data for both hands is a 6 value feature for each row and has the time dimension. This is also flattened to a 1-dimensional array to calculate Shannon Entropy. Projecting position data with high errors obtained by the double integral of accelerometer values to a higher dimension will increase errors and hence it is not performed.

The time-series data d_i is considered as input to calculate the Shannon Entropy $\sum_{i=1}^{n} -p(d_i)\log_2(p(d_i))$. The data is distributed into histograms and entropy is calculated as explained in Alg. 1.

The entropy of linear stick-based image model is similar to that of the non-linear hand model. The stick model lacks the information that a 3D non-linear hand model

Algorithm 1 Entropy Calculation

```
1: procedure SHANNON ENTROPY
2:     dataSet ← list of unique items in timeSeries
3:     freqList ← [ ]
4:     for entry in dataSet do
5:         counter ← 0
6:         for i in timeSeries do
7:             if i = entry then counter ← counter + 1.
8:             tsLen ← length of timeSeries
9:             freq ← counter/tsLen
10:            append freq to freqList
11:    ent ← 0.0
12:    for freq in freqList do
13:        ent ← ent+freq*log(freq)
14:    ent ← -ent
15:    return ent
```

contains, since this only uses pitch and yaw information. The lack of information affects the recognition accuracy and this is improved with the help of a 3D human non-linear hand model by incorporating the roll information.

4.2.5 Error Estimation

An error estimation is performed on the synthetically generated dataset to see how much trajectory shift occurs during the conversion. The palm of the 3D human hand model is tracked and the Cartesian coordinates in the XYZ space for both hands is obtained. For each ASL word instance, the hand positions are normalized

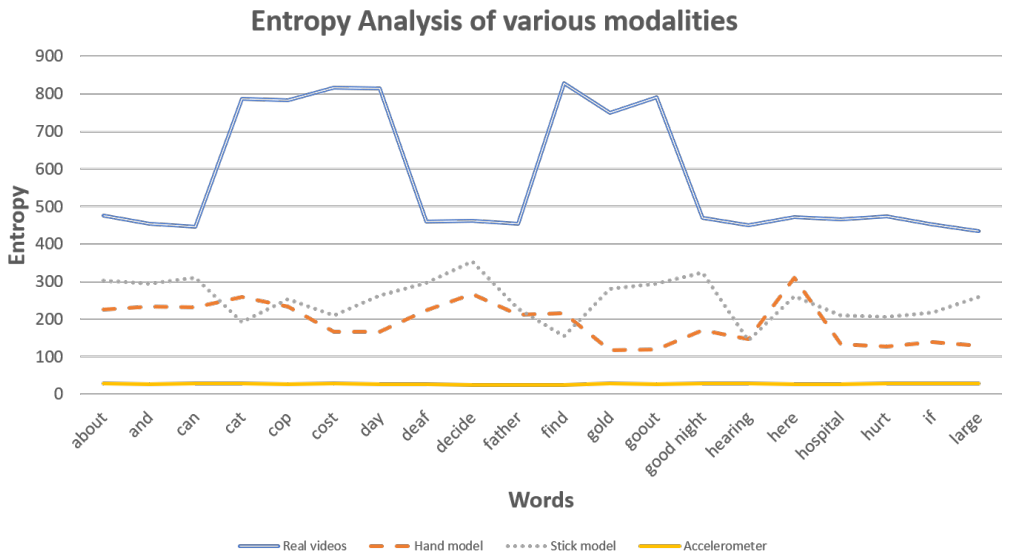

Figure 4.4: Entropy Analysis From Using Various Representations of Data.

between 0 and 1 for effective comparison with other instances. The actual XYZ coordinates of the palm for both hands is obtained from the Kinect which tracks the wrist joint. Root mean squared error is used to compute the average error for each ASL word and is averaged over all the users between the Myo generated(reconstruction) and Kinect(actual) coordinates. The RMSE normalized is shown in Fig. 4.5. Since only the movement and rotation of only the elbow joint is considered, the upper arm movement is ignored. However, many of the ASL signs involved significant upper arm movement and these words have higher error compared to the words which involve no upper arm movement. Minimizing this error would certainly improve the accuracy of the system.

Summary:

In this chapter, synthetic hand movement videos are generated and a detailed analysis of the generated videos consisting of error estimation and entropy analysis

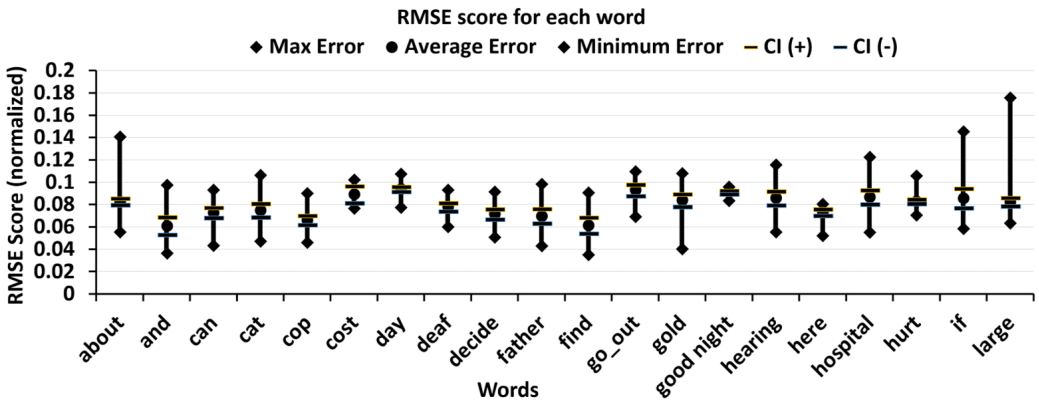

Figure 4.5: The Normalized RMSE Error of Converting Quaternions to Hand Model Compared to the Actual Kinect Ground Truth Position Coordinates.

is performed. The synthetic scene and camera can be varied for various viewpoint generations of the video. The entropy of the frame images is used to calculate the amount of information present in the videos and the gain in information is quantified by comparing with the entropy of raw signals. The depth sensor position value is considered as ground truth and an RMSE error estimation is performed to calculate the error in generation.

Chapter 5

MODEL ARCHITECTURES

In this chapter, the deep learning architectures used to perform recognition are explained in detail. Convolution 3D (C3D) [33] and the two stream networks from the Temporal Segment Networks [35] are used to test this hypothesis.

5.1 Convolution 3D

Convolution 3D (C3D) net has 8 convolution, 5 max-pooling, and 2 fully connected layers, followed by a soft-max output layer as shown in Fig. 5.1. All 3D convolution kernels are $3 \times 3 \times 3$ with stride 1 in both spatial and temporal dimensions. Number of filters are denoted in each box. The 3D pooling layers are denoted from pool1 to pool5. All pooling kernels are $2 \times 2 \times 2$, except for pool1 is $1 \times 2 \times 2$. Each fully connected layer has 4096 output units.

The same approach is followed as mentioned in the C3D paper. The feature extraction is as follows: The video is split into 16 frame long clips and has a 8-frame overlap between two consecutive clips. They are fed into the network and the fc6 activations are obtained. These activations are averaged and followed by a L2-norm for the final output.

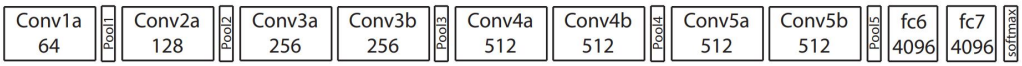

Figure 5.1: Convolution 3D Architecture [33].

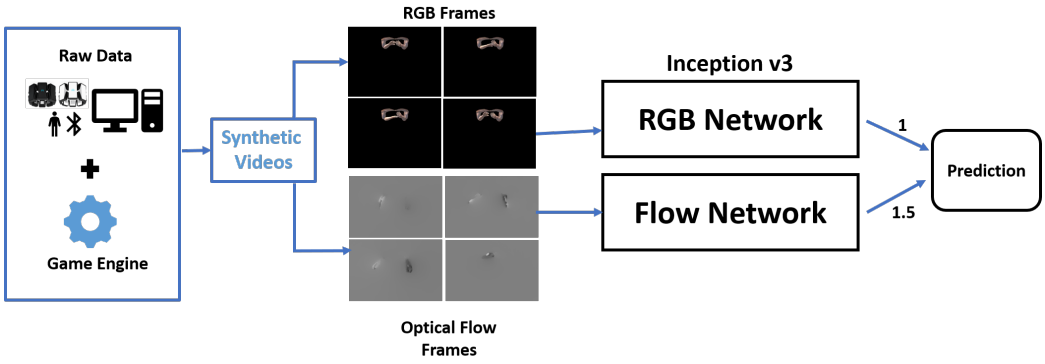

Figure 5.2: Two Stream Model Overview for Synthetic Video Recognition [35].

5.2 Two Stream Networks

The temporal segment networks are an improvisation on the Two Stream Model [30]. They use Inception architecture and an ImageNet [16] prior as initialization. Optical flow frames in addition to the RGB frames are generated, to train the Two Stream networks as shown in Fig. 5.2. For the RGB Network, the learning rate is initialized as 0.001 and decreases 0.1 every 1,500 iterations. The total number of iterations is 3,500. For the Flow network, initialize the learning rate as 0.005 and it reduces by 0.1 after 10,000 and 16,000 iterations. The total number of iterations is 18,000. The optical flow extraction technique is the same as used in the paper [35].

5.2.1 Optical Flow

Optical flow is used to calculate the displacement of brightness patterns between frames by using the information from the neighboring pixels [20]. All the pixels are considered when extracting optical flow here, hence, this is a dense optical flow extraction.

Summary:

The convolution 3D architecture is used because it can effectively model both spatial and temporal dimensions of a video. The Two Stream model which uses two tracks of recognition is used. The two tracks are the RGB frame and the optical flow track. It combines the results of both networks to perform improved recognition on the videos.

Chapter 6

RESULTS

In this chapter, the various analysis on the models and their results are discussed. The concept of user independent gesture recognition, multimodal gesture recognition are very important topics and there is a great deal of ongoing research in both wearable sensor domain and in computer vision. Some of the existing techniques use signal processing techniques and performing feature extraction on the raw data [22, 23] but are user dependent. In this proposed method of constructing the dataset, only the raw sensor values are used and the same hand model with data from multiple user is animated. This effectively enables us to capture one of the core meanings of the gesture i.e., the hand trajectory. All the experiments below are user independent evaluations. A split of 32 users for training and 27 users for testing is considered.

6.1 Wearable Sensor - Armband sensors

For IMU based experiment, the statistical feature set benchmarked features from Thomaz et al. [32] experiment is used. The sensors from the Myo armband consists of gyroscope, accelerometer, orientation and EMG which is a total of 34 sensors. The feature set consists of five statistical features: mean, variable, skewness ($\frac{\sum_{n=1}^{N}(x_n-x)^3}{(N-1)s^3}$), kurtosis ($\frac{\sum_{n=1}^{N}(x_n-x)^4}{(N-1)s^4}$), and Root-Mean-Square ($\frac{1}{N}\sum_{n=0}^{N-1}|X_n|^2$). Therefore, the feature size for one instance is 170 (5 feature $\times$ 34 sensor). Then, traditional machine learning techniques (supervised learning) such as Naive Bayes (NB), Random Forest (RF), Support Vector Machine (SVM), and Deep Neural Network (DNN) are applied. The DNN model has four or five hidden layers with 256 or 512 nodes for each layer. The activation function is ReLU, and the gradient descent optimization is ADaptive

Moment Estimation (ADAM) [15]. The results are shown in Tab. 6.1. This model trained on synthetic videos as discussed in Chapter 5 significantly outperforms the other models.

Table 6.1: Signal Processing Versus MirrorGen TwoStream Model.

Model	**34 sensors**	**Orientation (6 sensors)**
Naive Bayes	34.51	26.38
SVM	50.65	46.34
Random Forest	71.63	60.36
DNN (255 Nodes 4 Layers)	56.72	47.87
DNN (512 Nodes 5 Layers)	50.85	44.34
Proposed Method	**-**	**99.03**

6.2 Results on Convolution 3D

From the generated synthetic videos, for each gesture instance, a 16-frame non-overlapping set is used as the input to the network which is the same as the [33]. Only spatial images are used here and no preprocessing is required. The results on Convolution 3D is shown in Tab. 6.2. It is observed that the Convolution 3D effectively captures the salient motion of the arm.

The confusion matrix shown in Fig. 6.1 and in Fig. 6.2, is the average prediction scores for each ASL word from the 20 word S-ASL dataset. Words like cat, decide involve significant upper arm movement. This affects the accuracy of the model since only the lower arm movement is considered but the effect is not that significant. The

Table 6.2: Results Using Synthetic Videos on Convolution 3D.

Dataset	Accuracy
One Angle	88.68
Three Angle	92.75

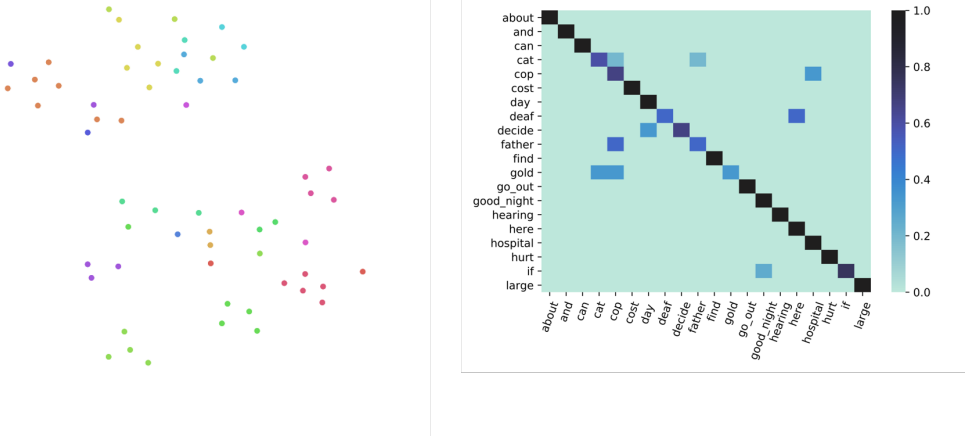

Figure 6.1: **Left:** t-SNE Visualization of the fc2 Layer of the Convolution 3D Trained Network on Direct Angle Dataset **Right:** Confusion Matrix Generated Based on Average Prediction Probabilities of Multiple Instance of Each Gesture From S-ASL Using C3D on Direct Angle Dataset.

minor glows in the confusion matrix are because of the semantic similarity between the signs. For example, the word "if" and "go out" have the same region of execution which is near the dominant arm shoulder.

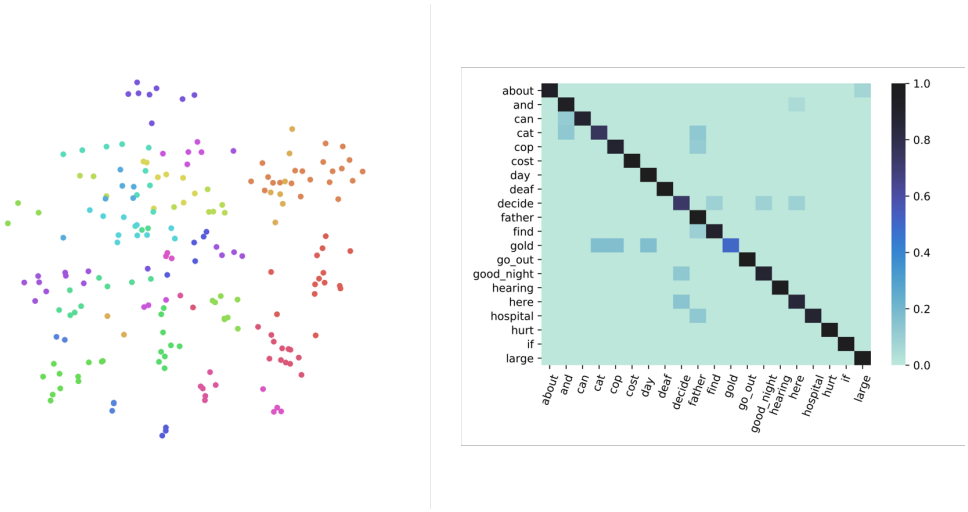

Figure 6.2: **Left:** t-SNE Visualization of the fc2 Layer of the Convolution 3D Trained Network on 3-Angle Dataset **Right:** Confusion Matrix Generated Based on Average Prediction Probabilities of Multiple Instance of Each Gesture From S-ASL Using C3D on 3-Angle Dataset.

6.3 Results on Temporal Segment Networks

6.3.1 Real - RGB Videos

Since RGB frames are also collected during the data collection process, training the same Two Stream network on only real RGB videos by extracting optical flow for RGB videos is also performed. These videos have a person standing in front of the camera and perform all the ASL words. The results for various size of training data is shown in increasing order in Tab. 6.3.

6.3.2 Synthetic Videos

The model is trained on synthetic videos from 32 train users and tested on synthetic videos from 27 test users. The results are shown in Tab. 6.4.

Both training and testing on generated video data is done to compare recognition

Table 6.3: Results for Train and Test on RGB Videos (No Synthetic Data) Using the Two Stream Model.

No of Real Videos	**RGB**	**Flow**	**Fusion**
2000	78.52	91.12	91.50
5000	80.16	91.53	91.46
10000	80.56	90.75	91.20

Table 6.4: Results for Train and Test on Synthetic Videos Using the Two Stream Model.

Dataset	**RGB**	**Flow**	**Fusion**
One Angle	84.46	**99.03**	99.02
Three Angle	80.31	98.98	98.81

accuracies of purely IMU based data with other recognition models as shown in the Tab. 6.1. The Flow model performs better since the dataset has more temporal information.

The confusion matrix and the t-SNE visualization [19] of the global-pooling layer for the optical flow model of the one-angle video generation is shown in Fig. 6.3. The clustering of data points in the t-SNE visualization shows effective learned representation of the global-pool layer of the network.

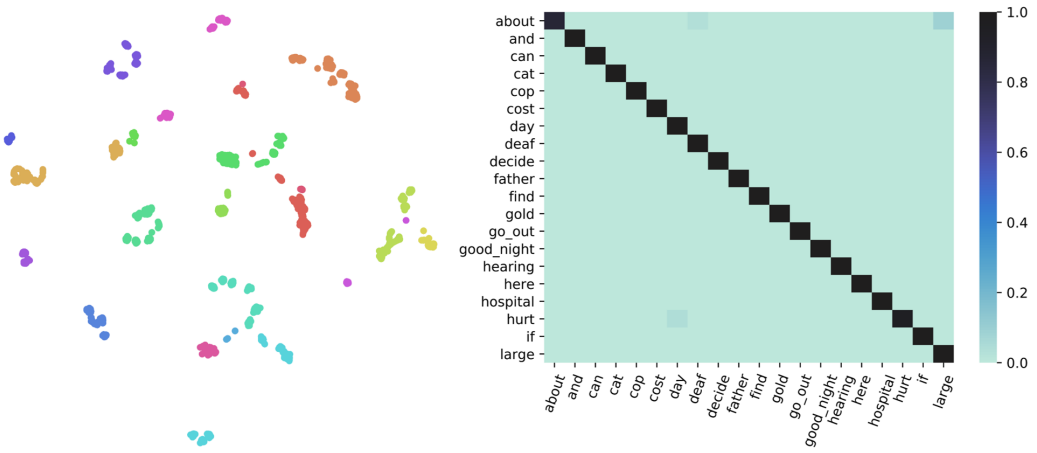

Figure 6.3: **Left:** t-SNE Visualization of the Global-Pooling Layer of the Optical Flow Trained Network Using Synthetic Data. **Right:** Confusion Matrix Generated Based on Normalized Average Prediction Score of Multiple Instances of Each Gesture From the Optical Flow Trained Network Using Synthetic Data.

6.4 Discussion

The objective here is to use minimal amount of real videos with a large amount of synthetic videos to get similar performance. The ratio is fixed at 80% generated videos - 20% real videos as the threshold point and further experimented by varying the size of the train data as shown in Tab. 6.5 and Fig. 6.5.

The architecture shown in Fig. 5.1 is modified to support mix of synthetic videos as shown in Fig. 6.4

We can see that from the 20-80 split there is a significant increase in accuracy. The threshold point is fixed at 20-80 and is further experimented by varying the size of the train data. The above experiment was done with a fixed train data size of 10,000 videos.

To evaluate the performance of synthetic videos, a comparison test between a model trained only on Real videos and a model trained on a mix of real and synthetic

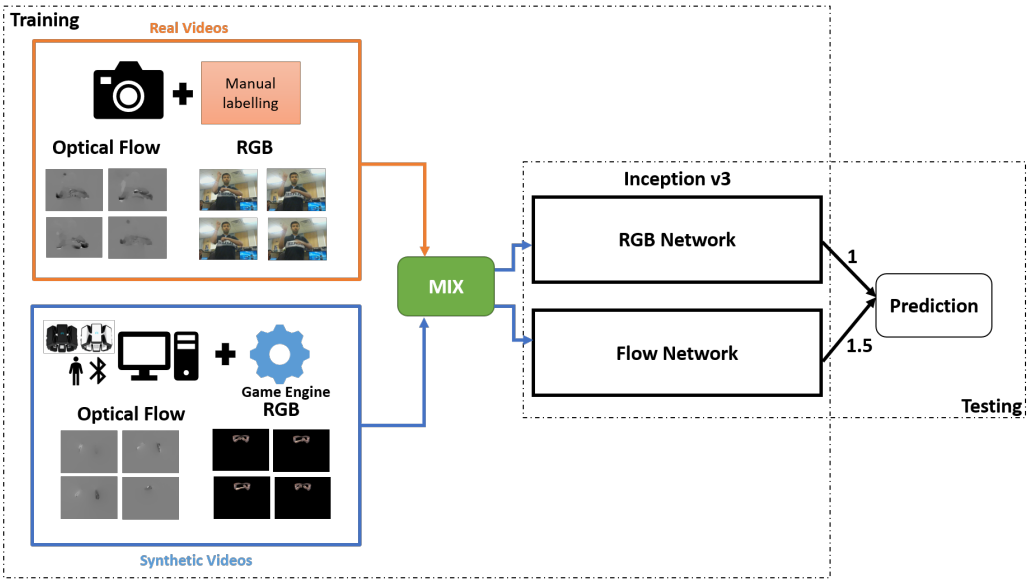

Figure 6.4: Modified System Overview to Support Mixing of Real and Synthetic Videos.

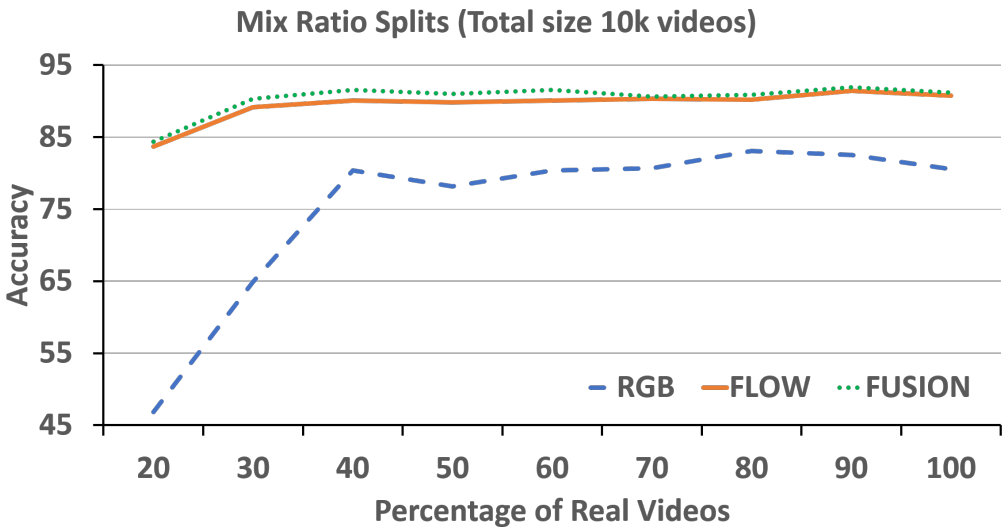

Figure 6.5: Change in Accuracy of the Flow Model Over Various Mix Ratios of Videos on a Total of 10000 Videos.

Table 6.5: Performance of the Two Stream Model Over Various Mix Ratios of Videos on a Total of 10000 Videos.

Train Split (%)		**Accuracy (%)**		
Real videos	**Synthetic videos**	**RGB**	**Flow**	**Fusion**
100	0	80.56	90.75	91.20
90	10	82.54	91.43	91.91
80	20	80.39	90.09	91.58
70	30	78.18	89.82	91.01
60	40	83.09	90.23	90.88
50	50	80.70	90.30	90.66
40	60	75.64	88.04	89.56
30	70	64.92	89.19	90.31
20	80	46.87	83.72	84.36

videos are tested on a test dataset containing both real and synthetic videos. The mix model performs significantly better than the model trained only on real videos is shown in Fig. 6.6.

The confusion matrix and the tSNE visualization of the global-pooling layer for the optical flow model of the 90 real - 10 generated model with highest accuracy of 91.91% video generation is shown in Fig. 6.7. The clustering of data points in the tSNE visualization shows effective learned representation of the global-pool layer.

The average entropy for all the modalities is calculated by averaging the entropy of all the words. As shown in Tab. 6.6, the RGB videos have information unnecessary for gesture recognition. They have a high entropy which affects the accuracy. Prepro-

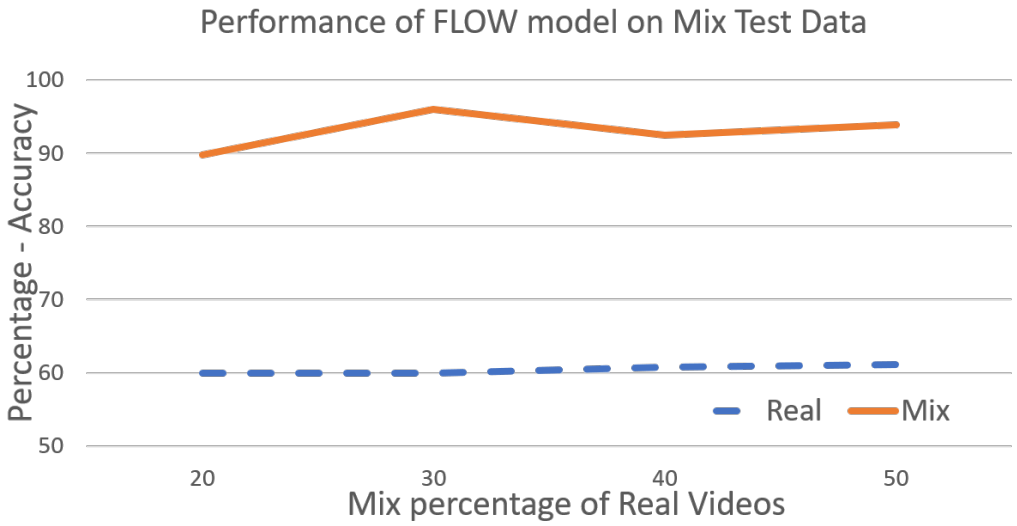

Figure 6.6: Accuracy of the Flow Model Versus Splits Ratio Graph Containing Both Real and Mix Model Data on a Total of 10000 Videos.

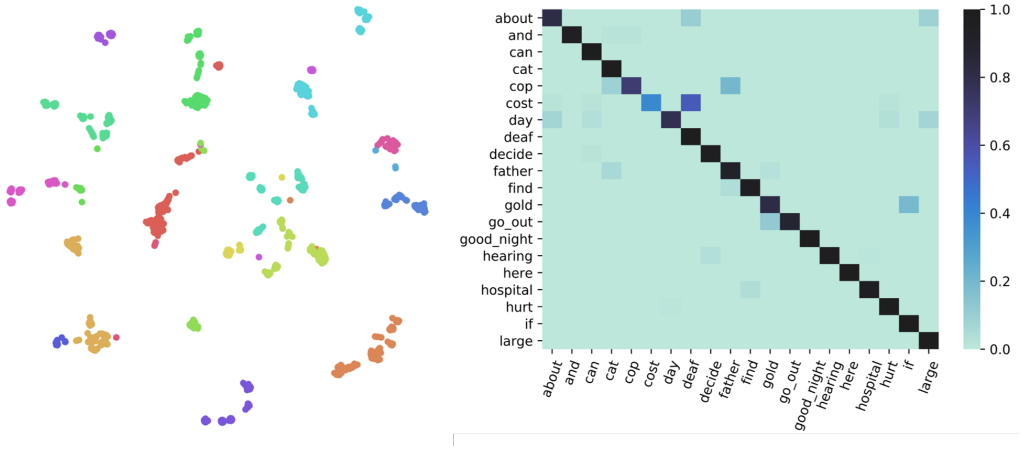

Figure 6.7: t-SNE Visualization of the Global-Pooling Layer of the Inception Optical Flow Trained Network for 9010 Mix Model. Confusion Matrix Generated Based on Average Prediction Probabilities of Multiple Instances of Each Gesture From S-ASL Using the 9010 Mix Model.

cessing can be performed to remove background and perform recognition, however, this is not feasible in real-time and significant human effort is needed to perform guided preprocessing. The stick model even though has a similar entropy to the hand model lacks in roll information. When a model trained on synthetic hands is tested on RGB videos (Real) they give very low accuracy since the distribution of data is very different. Since, the non-linear hand model gives a low accuracy the stick model will perform equally bad, hence, this experiment is not performed. For the Mix model, the best performing mix model (90-10 mix) is considered. The mix model performs marginally better compared to the all real model but still has high entropy. The entropy of the mixed model is a weighted average of the two modalities (90-10 mix).

Table 6.6: Comparison of Real, Hand and Stick Models Accuracies.

Modality	Real	Stick	Hand	Mix (Hand & Real)
Average Entropy (bits)	577.08	191.17	252.82	544.65
Real - Train	91.20	-	-	-
Stick - Train	-	93.75	-	-
Hand - Train	10.67	-	99.03	-
Mix (Hand & Real)	91.91	-	99.03	92.62

Summary: The results based purely on using the raw sensor data using signal processing techniques for feature extraction is discussed first and compared with the proposed thesis work. Summary on the experiments on the two architectures are as follows: The two stream model is found to perform the best on synthetic data. Further, a recognition model purely on real video based input instead of synthetic

input is also used. The generated synthetic videos can be used to augment the real videos for improving results as shown in the discussion section. An analysis of various modalities to intuitively infer the gain in information is performed.

Chapter 7

CONCLUSIONS

In this thesis, a novel way of generating synthetic video dataset for hand gestures by focusing mainly on the rotation of the lower arm is discussed. There is a significant accuracy increase for recognition using synthetic video vision as compared to conventional signal processing. This is partly because robust pre-trained mid-level features are available for video recognition but are not available for raw sensor signals.

The idea of using this generated synthetic videos to improve recognition accuracy on real-world gesture datasets is experimented. The experimental results for mixing synthetic and real-world data for training to test on real-world videos is shown to support the idea of data augmentation using MirrorGen generated videos.

By using synthetically generated videos, it can be seen that they can be used as a substitute for problems where the actual dataset is small. The small dataset can be augmented with synthetic videos to make a large scale dataset for training large networks without affecting the accuracy and to perform recognition in different modalities.

In addition to performing well on the transformed sensor data, this synthetic dataset also helps in multitask classification of real-time gesture data by serving as a relevant prior.

Future Work:

(a) The wearable sensors used also provide EMG data for the lower arm. This EMG data can be further used to improve recognition confidence by differentiating similar hand rotations which have different wrist/palm poses.

(b) The usage of synthetic video to augment the training data is analyzed. However, they do not show promising consistent increase and there is marginal gain in some specific cases. This is because of the difference in distribution between the synthetic videos and the real videos. This distribution gap problem could be solved by introducing a whole body avatar instead of only synthetic arms.

(c) Further, this proposed method can also be used for recognition of generic hand gestures (pantomimes). This may involve detecting gestures like opening a door, waving hands and even continuous gestures involving complex activities like eating [18] and drinking.

(d) Using information from additional sensors other than armband sensors like a smartwatch, leap motion sensor or ear mounted sensors [3], more information can be added to the generated synthetic videos to recognize gestures which also involve parts other than the lower arm.

(e) Using wearable sensors to interact with Virtual Reality applications coupled with gesture recognition opens the door to countless opportunities since multiple combinations of gestures are possible with high precision recognition using synthetic models.

REFERENCES

[1] Adelstein, F., S. K. Gupta, G. Richard and L. Schwiebert, *Fundamentals of mobile and pervasive computing*, vol. 1 (McGraw-Hill New York, 2005).

[2] Ballan, L., A. Taneja, J. Gall, L. Van Gool and M. Pollefeys, "Motion capture of hands in action using discriminative salient points", in "European Conference on Computer Vision", pp. 640–653 (Springer, 2012).

[3] Bedri, A., R. Li, M. Haynes, R. P. Kosaraju, I. Grover, T. Prioleau, M. Y. Beh, M. Goel, T. Starner and G. Abowd, "Earbit: using wearable sensors to detect eating episodes in unconstrained environments", Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies **1**, 3, 37 (2017).

[4] Camgoz, N. C., S. Hadfield, O. Koller and R. Bowden, "Subunets: End-to-end hand shape and continuous sign language recognition", in "IEEE International Conference on Computer Vision (ICCV)", (2017).

[5] Castrejon, L., Y. Aytar, C. Vondrick, H. Pirsiavash and A. Torralba, "Learning aligned cross-modal representations from weakly aligned data", arXiv preprint arXiv:1607.07295 (2016).

[6] Chung, J., J. Chung, W. Oh, Y. Yoo, W. G. Lee and H. Bang, "A glasses-type wearable device for monitoring the patterns of food intake and facial activity", in "Scientific reports", (2017).

[7] Cui, R., H. Liu and C. Zhang, "Recurrent convolutional neural networks for continuous sign language recognition by staged optimization", 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR) pp. 1610–1618 (2017).

[8] de Souza, C. R., A. Gaidon, Y. Cabon and A. L. Pena, "Procedural generation of videos to train deep action recognition networks", in "Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)", vol. 2 (2017).

[9] Dong, C., M. C. Leu and Z. Yin, "American sign language alphabet recognition using microsoft kinect", in "Computer Vision and Pattern Recognition Workshops (CVPRW), 2015 IEEE Conference on", pp. 44–52 (IEEE, 2015).

[10] Emmorey, K., J. Xu, P. Gannon, S. Goldin-Meadow and A. Braun, "Cns activation and regional connectivity during pantomime observation: No engagement of the mirror neuron system for deaf signers", Neuroimage **49**, 1, 994–1005 (2010).

[11] Fang, B., J. Co and M. Zhang, "Deepasl: Enabling ubiquitous and non-intrusive word and sentence-level sign language translation", in "Proceedings of the 15th ACM Conference on Embedded Network Sensor Systems", p. 5 (ACM, 2017).

[12] Gaidon, A., Q. Wang, Y. Cabon and E. Vig, "Virtual worlds as proxy for multi-object tracking analysis", arXiv preprint arXiv:1605.06457 (2016).

[13] Hoffman, J., S. Gupta and T. Darrell, "Learning with side information through modality hallucination", in "Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition", pp. 826–834 (2016).

[14] Karpathy, A., G. Toderici, S. Shetty, T. Leung, R. Sukthankar and L. Fei-Fei, "Large-scale video classification with convolutional neural networks", in "Proceedings of the IEEE conference on Computer Vision and Pattern Recognition", pp. 1725–1732 (2014).

[15] Kingma, D. P. and J. Ba, "Adam: A method for stochastic optimization", arXiv preprint arXiv:1412.6980 (2014).

[16] Krizhevsky, A., I. Sutskever and G. E. Hinton, "Imagenet classification with deep convolutional neural networks", in "Advances in neural information processing systems", pp. 1097–1105 (2012).

[17] Lawrence, S., C. L. Giles, A. C. Tsoi and A. D. Back, "Face recognition: A convolutional neural-network approach", IEEE transactions on neural networks **8**, 1, 98–113 (1997).

[18] Lee, J., P. Paudyal, A. Banerjee and S. K. Gupta, "Fit-eve&adam: Estimation of velocity & energy for automated diet activity monitoring", in "Machine Learning and Applications (ICMLA), 2017 16th IEEE International Conference on", pp. 1071–1074 (IEEE, 2017).

[19] Maaten, L. v. d. and G. Hinton, "Visualizing data using t-sne", Journal of machine learning research **9**, Nov, 2579–2605 (2008).

[20] Nourani-Vatani, N., P. V. Borges and J. M. Roberts, "A study of feature extraction algorithms for optical flow tracking", in "Australasian Conference on Robotics and Automation", (2012).

[21] Oquab, M., L. Bottou, I. Laptev and J. Sivic, "Learning and transferring mid-level image representations using convolutional neural networks", in "Computer Vision and Pattern Recognition (CVPR), 2014 IEEE Conference on", pp. 1717–1724 (IEEE, 2014).

[22] Paudyal, P., A. Banerjee and S. K. Gupta, "Sceptre: a pervasive, non-invasive, and programmable gesture recognition technology", in "Proceedings of the 21st International Conference on Intelligent User Interfaces", pp. 282–293 (ACM, 2016).

[23] Paudyal, P., J. Lee, A. Banerjee and S. K. Gupta, "Dyfav: Dynamic feature selection and voting for real-time recognition of fingerspelled alphabet using wearables", in "Proceedings of the 22nd International Conference on Intelligent User Interfaces", pp. 457–467 (ACM, 2017).

[24] Rogez, G. and C. Schmid, "Image-based synthesis for deep 3d human pose estimation", arXiv preprint arXiv:1802.04216 (2018).

[25] Saenko, K., B. Kulis, M. Fritz and T. Darrell, "Adapting visual category models to new domains", in "European conference on computer vision", pp. 213–226 (Springer, 2010).

[26] Sandler, W., "The phonological organization of sign languages", Language and linguistics compass **6**, 3, 162–182 (2012).

[27] Savur, C. and F. Sahin, "American sign language recognition system by using surface emg signal", in "Systems, Man, and Cybernetics (SMC), 2016 IEEE International Conference on", pp. 002872–002877 (IEEE, 2016).

[28] Sigurdsson, G. A., S. Divvala, A. Farhadi and A. Gupta, "Asynchronous temporal fields for action recognition", in "CVPR", vol. 2, p. 6 (2017).

[29] Sigurdsson, G. A., G. Varol, X. Wang, A. Farhadi, I. Laptev and A. Gupta, "Hollywood in homes: Crowdsourcing data collection for activity understanding", in "European Conference on Computer Vision", pp. 510–526 (Springer, 2016).

[30] Simonyan, K. and A. Zisserman, "Two-stream convolutional networks for action recognition in videos", in "Advances in neural information processing systems", pp. 568–576 (2014).

[31] Soomro, K., A. R. Zamir and M. Shah, "Ucf101: A dataset of 101 human actions classes from videos in the wild", arXiv preprint arXiv:1212.0402 (2012).

[32] Thomaz, E., I. Essa and G. D. Abowd, "A practical approach for recognizing eating moments with wrist-mounted inertial sensing", in "Proceedings of the 2015 ACM International Joint Conference on Pervasive and Ubiquitous Computing", pp. 1029–1040 (ACM, 2015).

[33] Tran, D., L. Bourdev, R. Fergus, L. Torresani and M. Paluri, "Learning spatiotemporal features with 3d convolutional networks", in "Computer Vision (ICCV), 2015 IEEE International Conference on", pp. 4489–4497 (IEEE, 2015).

[34] Vapnik, V. and A. Vashist, "A new learning paradigm: Learning using privileged information", Neural networks **22**, 5-6, 544–557 (2009).

[35] Wang, L., Y. Xiong, Z. Wang, Y. Qiao, D. Lin, X. Tang and L. Van Gool, "Temporal segment networks: Towards good practices for deep action recognition", in "European Conference on Computer Vision", pp. 20–36 (Springer, 2016).

[36] Wang, X., A. Farhadi and A. Gupta, "Actions˜ transformations", in "Proceedings of the IEEE conference on Computer Vision and Pattern Recognition", pp. 2658–2667 (2016).

[37] Wang, X. and A. Gupta, "Unsupervised learning of visual representations using videos", arXiv preprint arXiv:1505.00687 (2015).

[38] Yun, L. K., T. T. Swee, R. Anuar, Z. Yahya, A. Yahya and M. R. A. Kadir, *Sign Language Recognition System Using SEMG and Hidden Markov Models*, Ph.D. thesis, Universiti Teknologi Malaysia (2012).

[39] Zitnick, C. L. and D. Parikh, "Bringing semantics into focus using visual abstraction", in "Computer Vision and Pattern Recognition (CVPR), 2013 IEEE Conference on", pp. 3009–3016 (IEEE, 2013).