Generation-Based Data Augmentation Pipeline for Real-Time Automatic Gesture Recognition

Mingxi Cheng[1a][b], Fatima Zohra Daha[1a][c], Amit Srivastava[2] and Ji Li[1][d]
[1]*Microsoft, Mountain View, CA, U.S.A.*
[2]*ServiceNow, Santa Clara, CA, U.S.A.*

Keywords: Gesture Recognition, GAN, 3D-CNN, Deep Learning.

Abstract: With the SARS-CoV-2 pandemic outbreak, video conferencing tools experience huge spikes in usage. Gesture recognition can automatically translate non-verbal gestures into emoji reactions in these tools, making it easier for participants to express themselves. Nonetheless, certain rare gestures may trigger false alarms, and acquiring data for these negative classes in a timely manner is challenging. In this work, we develop a low-cost fast-to-market generation-based approach to effectively reduce the false alarm rate for any identified negative gesture. The proposed pipeline is comprised of data augmentation via generative adversarial networks, automatic gesture alignment, and model retraining with synthetic data. We evaluated our approach on a 3D-CNN based real-time gesture recognition system at a large software company. Experimental results demonstrate that the proposed approach can effectively reduce false alarm rate while maintaining similar accuracy on positive gestures.

1 INTRODUCTION

Speakers in all cultures gesture when they talk (Goldin-Meadow and Alibali, 2013). Even congenitally blind individuals, who have never seen any gesture, move their hands when they talk, highlighting the universality and robustness of hand gesture in communication (Iverson and Goldin-Meadow, 1998; Goldin-Meadow and Alibali, 2013). Accordingly, automatic hand gesture recognition becomes an important computer vision research topic with considerable downstream applications, such as human-computer interaction, sign language interpretation, touchless interfaces and non-verbal communication systems (Min et al., 2020; Liu et al., 2020).

With the SARS-CoV-2 pandemic outbreak, it becomes particularly interesting to incorporate hand gesture recognition into video conference software, such as Microsoft Teams (Microsoft,), Cisco WebEx (Cisco Systems,), Google Meet (Google,), and Zoom (Zoom,), to support automatic reactions for enhanced remote social touch. The model can be hosted either in the cloud, such as Microsoft Azure (azu,), Google Cloud (goo,), and Amazon Web Services (aws,), or on users' end devices. There are mature production pipelines to deploy the model candidates in the cloud for better maintainability. However, from an economic point of view, hosting models on users' end devices is becoming more and more attractive as compared to the high cost of hosting models in the cloud where cloud service providers have to pay for the electric bills of data centers.

Despite the economic benefit, serving real-time hand gesture recognition models faces severe challenges. Firstly, the model must be small enough to be deployable to majority of the edge devices, which could be a smartphone, a desktop, or a web browser. This ensures consistent user experience across different edge platforms. Secondly, the model must be fast enough to achieve real-time response in order to provide smooth auto-reactions in a video conference. Thirdly, the model must be accurate enough to avoid any misfire so as to avoid bringing embarrassment to the online video meeting.

There are numerous techniques proposed to address the aforementioned small model size and low latency challenges, such as data dependent or independent efficient architectures (Zaheer et al., 2020; Tay et al., 2020), parallel computing (Rajbhandari et al., 2020), memory optimization (Pudipeddi et al., 2020), and model compression techniques, such as knowledge

[a]These authors contributed equally to this work.
[b] https://orcid.org/0000-0002-8070-6665
[c] https://orcid.org/0000-0002-1142-0874
[d] https://orcid.org/0000-0003-4699-084X

Cheng, M., Daha, F., Srivastava, A. and Li, J.
Generation-Based Data Augmentation Pipeline for Real-Time Automatic Gesture Recognition.
DOI: 10.5220/0011886600003393
In *Proceedings of the 15th International Conference on Agents and Artificial Intelligence (ICAART 2023) - Volume 2*, pages 437-446
ISBN: 978-989-758-623-1; ISSN: 2184-433X

distillation (Hinton et al., 2015), quantization (Zafrir et al., 2019), and pruning (Brix et al., 2020).

The high precision requirement, on the other hand, turns out to be more challenging in practice. Many prior arts in the field of hand gesture recognition reported high accuracy numbers on various benchmarks. Nonetheless, in subjective evaluations with a group of users, we find that model candidates with high accuracy scores on benchmarks can have high false alarm rates for gestures which are not included in the training data. For example, a model candidate trained for recognizing "hand raise" and "thumb up" can get confused by "quote sign" and "peace sign", which are not included in the training data. Getting sufficient amount of data for these negative classes is difficult, as the entire process of data preparation, labeling, verification, and legal review costs extra time and money.

The majority of recent research efforts in video-based gesture recognition is focused on 3D hand tracking (Boukhayma et al., 2019; Ge et al., 2019; Mueller et al., 2018), skeleton/pose estimation (Iqbal et al., 2018; Doosti et al., 2020; Liu et al., 2020; Min et al., 2020), and accuracy improvement (Narayana et al., 2018; Abavisani et al., 2019; Tang et al., 2019; Chen et al., 2019). There is a serious lack of research of effective approaches to reduce false alarm rate for video-based hand gesture recognition, which is critical to its success in video conference software.

In this work, we develop a low-cost fast-to-market generation-based approach to effectively reduce the false alarm rate of hand gesture recognition for any target negative class. We propose a pipeline starting from data augmentation via generative adversarial networks (GANs), and followed with an alignment algorithm to further improve the generation results. To address the false alarm issue, we must take into consideration a diverse training set, where deep learning models can learn representative and diverse patterns and gain generalization ability to achieve good performance in real world applications or external independent test sets. Our proposed pipeline learns to take existing training data and insert negative gestures to produce useful samples, which preserve visual features from original training set, and provide enough information about negative classes. We find that the generated synthetic data helps deep learning models gain abilities to recognize unseen negative gestures and distinguish them from pre-defined positive classes, even when it does not look perfect by human inspection. We evaluate our augmentation pipeline with the proposed 3D-Convolutional Neural Network (3D-CNN) model and a pre-trained real-time vision model, MobileNet, to demonstrate its superiority in terms of false alarm reduction.

The contributions of this work are as follows:

- We propose a 3D-CNN model for automatic reaction function for online meeting applications, where the 3D-CNN model maps video streams to hand gesture reactions accurately in real-time.
- We propose a data augmentation pipeline to enrich training set by generating synthetic hand gesture data, which preserves the original visual features from real world data and includes unseen hand gestures. With augmented datasets, we improve our proposed 3D-CNN model and a bench-marking model in both internal and external test sets.
- We utilize an alignment algorithm to further improve the quality of our generated synthetic data, and we demonstrate empirically that our pipeline significantly reduces false alarm rate while achieving equivalent or higher accuracy performance.

The rest of this paper is organized as follows. In Section 2, we provide a overview of related work including data augmentation and hand gesture recognition. In Section 3, we introduce the proposed 3D-CNN gesture recognition model, our GAN-based data augmentation pipeline and an alignment algorithm to improve generation quality. In Section 4, we describe the datasets we used in our experiments and training details. We also include a ablation study to elaborate the effectiveness of our proposed pipeline. Section 5 provides a discussion and concludes the paper.

2 RELATED WORK

One of the challenges in real-time hand gesture recognition in online meeting environment is the trade-off between the accuracy and false alarms. Models that are sensitive to certain pre-defined positive classes usually are prone to false alarms, meaning that negative gestures also triggers automatic reactions, e.g., a random hand wave can trigger "raise hand" reaction. As seen in other deep learning tasks, training or fine-tuning models with known and unknown negative gestures contributes to false alarm rate reduction. Data augmentation methods, such as collecting a new dataset with negative classes, augmenting existing data with image processing techniques, and generating synthetic videos based on current dataset, could enrich the negative class in training data, and hence help reduce false alarms.

Synthetic Data Generation. Given the limited budget of collecting/purchasing new datasets, synthetic data generation emerges to be a preferable choice. Generative models, such as generative adversarial networks (GANs) and variational autoencoders

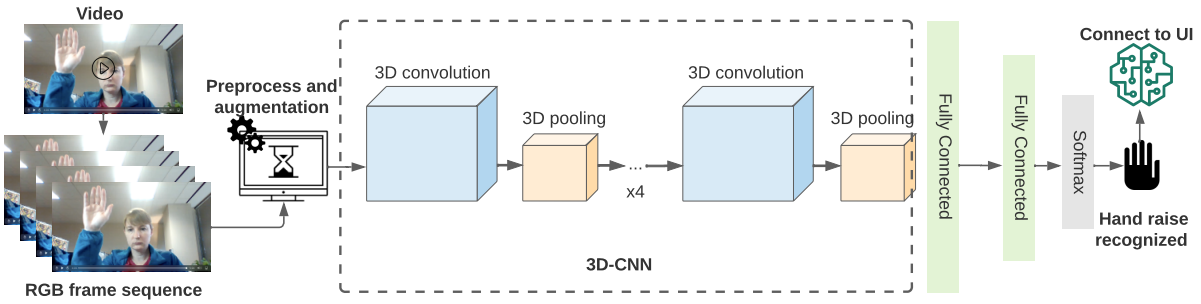

Figure 1: 3D-CNN architecture for hand gesture recognition. We take four central consecutive frames from the input video and preprocess them with our proposed augmentation pipeline (which will be discussed in the following sections). We then use augmented and preprocessed data to train the 3D-CNN model. The trained model then predicts a class and connect to down stream software in online meeting applications.

(VAEs), have shown to have promising results in various generation applications, e.g., image generation (Radford et al., 2015; Karras et al., 2017; Kingma and Welling, 2013; Rezende et al., 2014). Original GAN models generate realistic results in an unconditional fashion (Goodfellow et al., 2014), and later on, conditional GAN models are shown to have better and controllable results (Mirza and Osindero, 2014). Latest advanced GAN models that focusing on image to image translation, or style transfer have shown to be one of the best performing synthetic data generation methods. Pix2Pix (Isola et al., 2017) proposed by Isola et al. in 2017 is a not application specific image-to-image translation GAN framework. For instance, Pix2Pix can generate colorized photos from black and white images or even from sketches. The paired input-output is one of the key components behind the success of Pix2Pix. However, paired data is not always freely available in most of the application areas. CycleGAN (Zhu et al., 2017) developed in 2020, breaks the limitation of requiring paired input and successfully learns various image-to-image translation tasks with unpaired examples. Although image-to-image translation GAN works usually can be applicable to various domains, research work on controllable hand gesture generation or style transfer remains sparse. GestureGAN (Tang et al., 2020) proposed by Tang et al. is a keypoint/skeleton guided image-to-image translation framework which can generate realistic images with designated gestures. In this work, we exploit GestureGAN for our synthetic negative gesture augmentation.

Hand Gesture Recognition. Researchers have used different approaches to solve the hand gesture recognition task. Most of these approaches operate under specific constraints or require special hardware or gloves. In (Poon et al., 2019), Poon et al. proposed a bimanual hand gesture recognition technique where they fit independent SVM classifiers on the shape and color-encoded features. Recently, deep learning methods based on Recurrent Neural Network (RNN) and Convolutional Neural Network (CNN) have been extensively used and achieved remarkable results in hand gesture recognition. In (Karpathy et al., 2014), Karpathy et al. applied CNNs to extract spatial features from individual frames and fuse the temporal information. They explored different approaches for fusing information over temporal dimension through the network and concluded that the slow fusion method can get more global information in both spatial and temporal dimensions. In (Simonyan and Zisserman, 2014), Simonyan et al. proposed a two-stream ConvNet architecture which incorporates spatial and temporal networks. They capture the complementary information on appearance from still frames and motion between frames using optical flow. (Wang et al., 2015; Zhang et al., 2016; 139, 2016) extend the two-stream networks by integrating improved trajectories, motion vector and motion history image, respectively. In (Köpüklü et al., 2018), Köpüklü et al. involve both the motion data in the optical flow and the RGB frames. They fuse the information at the data level, then they adapt the pre-trained inception architecture. However, these methods require heavy preprocessing steps which make them unsuitable for real-time hand gesture detection. In this paper, we propose the use of a light-weight 3D-CNN to learn both spatial and temporal features. Then we apply static quantization to reduce the model size and latency.

3 PROPOSED METHOD

In this work, we propose a 3D-CNN model along with GAN-based data augmentation pipeline for automatic reaction function for online meeting applications. The design of 3D-CNN ensures real-time accurate hand gesture recognition, and when trained with data augmentation pipeline produced data, it achieves accurate recognition with low false alarm rates when tested in

online meeting setup. To further improve the quality of generated data, we develop an alignment algorithm to align input images and goal gesture images before sending these pairs to the GAN model. In this section, we first introduce the proposed 3D-CNN model in Section 3.1, then we describe our GAN-based data augmentation pipeline and alignment algorithm in Section 3.2 and Section 3.3.

3.1 3 D-CNN

We propose the use of 3D-CNNs for hand gesture recognition to capture both spatial and temporal information from video frames. The input of this model is a sequence of RGB frames captured from the user's camera feed. We also employ data augmentation techniques for a more effective training and to reduce potential overfitting. Figure 1 illustrates our approach which consists of three main phases: video preprocessing, feature learning, and classification.

The video processing step consists of converting the input video into RGB frames sequences, we take four successive frames as input and resize each frame to:

$$(w,h) = (\frac{width * 100}{height}, 100)$$

The input size is variable to assure the processing of videos with different aspect ratio (e.g., mobile, desktop). The RGB channels of each frame are also normalized. The resizing and normalization reduce the computation cost and training convergence of the model.

Our model consists of six 3D convolution layers and two fully connected layers followed by an adaptive average pooling layers to handle frames of different input sizes. The features extracted are then fed into a *softmax* layer for classification which outputs the probability of each class. To avoid overfitting and enhance the model generalization on test data, we apply dropout after the fully connected layer.

3.2 Augmentation

As shown in Figure 1, before the 3D-CNN takes RGB frame sequences as input and generates hand gesture recognition results, an augmentation step is performed to insert more diversity to data and improve model performance.

Data Processing. Data processing is an important part of the training phase. First, we duplicate the data by applying horizontal flipping to gestures that require one hand only. This step ensures that the training set is inclusive of both left and right hands. Furthermore, to ensure the diversity of our dataset, we apply lighting augmentation techniques, small translation and rotation. Moreover, to account for people who are far away from the camera, we apply padding. These image processing techniques are shown to be effective in data augmentation for computer vision tasks. To further tackle the false alarm issue in gesture recognition, we generate synthetic hand gesture data by inserting negative gestures into data with positive gestures. This brings the data augmentation into the next level and we will introduce it in the following section.

Synthetic Data Generation. In this work, we exploit the power of GAN and utilize GestureGAN (Tang et al., 2020) to generate synthetic data. GestureGAN is a keypoint/skeleton-guided controllable image to image translation method which takes pairs of image and skeleton and generates a image with the target gesture indicated in the skeleton image. It extends the idea of conditional GAN and CycleGAN (Zhu et al., 2017) and achieves outstanding generation results particularly in hand gesture translation.

Thus, we adopt and extend this controllable image to image translation to generate video frames containing the goal negative gestures while preserving the visual features extracted from source image frames as shown in Figure 2. Given a seed (source) video with any positive gesture, our goal is to replace the hand gesture in the seed video with the designated goal gesture, by utilizing a controllable image to image translation model, GestureGAN. For example, input image I contains finger raise gesture (g_{in}), and the designated output gesture g_{out} is peace sign, then the GAN generated image as shown in Figure 2 contains peace sign and preserves the visual features from I. The generation process is as follows.

- **Step 1.** Pre-process the seed video and extract the central f frames with the original positive gesture, e.g., f image frames containing finger raise gesture g_{in}, each of the frames is notated as I. Identify goal gesture s and generate a skeleton image $\mathcal{K}$ of the goal gesture, e.g., peace sign.
- **Step 2.** Detect hand gesture positions in the f seed frames and detect the skeleton position in the goal gesture image.
- **Step 3.** Utilize image processing techniques such as shrink, padding, and crop, to align f seed frames and the skeleton image $\mathcal{K}$ according to the hand/skeleton relative position.
- **Step 4.** Use GestureGAN to generate f goal frames with the goal gesture, e.g., peace sign, while preserving the visual features of seed frames.

The first step into the generation using GestureGAN is to pre-process the videos into image frames since GestureGAN is a image-based model. From our gen-

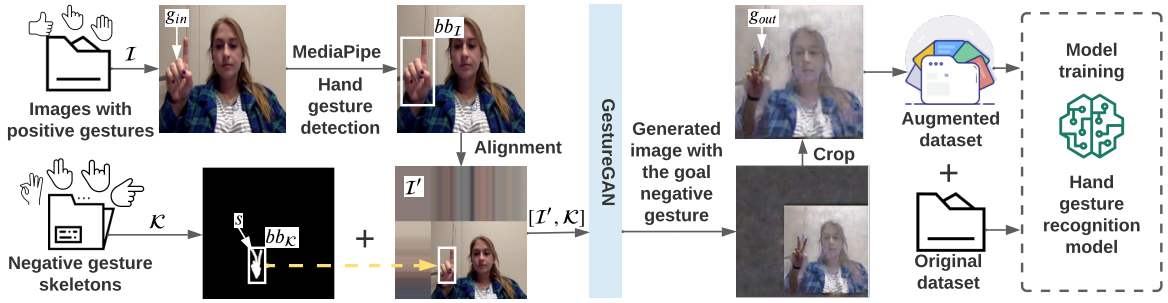

Figure 2: Synthetic data generation and alignment. With images containing positive gestures, e.g., finger raise, we generate synthetic data preserving the visual features from original images and containing designated negative gestures, e.g., peace sign. We first detect hand gesture in input image I and detect skeleton s in skeleton image $\mathcal{K}$, then based on the position and size of the detected hand and skeleton, we align the input image to make the hand gesture in it has similar size and position as s. We denote the processed input image as I' and GestureGAN takes the input pair $[I', \mathcal{K}]$ and generate an image containing the goal gesture, peace sign. The generated images are then processed and added into augmented dataset for later model training.

eration experience, we find that raw generation with random input pairs of image and skeleton, i.e., $[I, \mathcal{K}]$, leads to unfavorable results, which is caused by the mismatch of gesture positions between the input image and input skeleton. Therefore, we propose to detect hand gestures (g_{in}) and hand skeletons (s) in the input pairs (Step 2) and then align the input pairs according to the gesture-skeleton position (Step 3) before feeding into the generation model. To make our framework more easy to transfer, we utilize an open sourced hand detection package MediaPipe (Lugaresi et al., 2019) in Step 2. Note that any hand detection model that produces bounding boxes ($bb = [x^0, y^0, x^1, y^1]$, where x^0, y^0, x^1, y^1 represent top left and bottom right landmarks of the bounding box) for hands is applicable in this step. We introduce in detail our alignment algorithm in Section 3.3. Step 4 produces an augmented set, which we denote as $\mathcal{D}^*$.

3.3 Alignment

In this section, we introduce our alignment algorithm in detail. Assume we have a input pair, $[I, \mathcal{K}]$, with a image containing a original hand gesture g_{in}, and a skeleton image containing a goal gesture skeleton s. The alignment algorithm is shown in Algorithm 1. Note that this algorithm utilizes image processing methods to modify input image I. We can also process skeleton image $\mathcal{K}$, which we find to provide similar results in the end for generation. Once we have the aligned image-skeleton pair $[I', \mathcal{K}]$[1], we feed GestureGAN with these processed input pairs and generate image frames containing similar visual features as I' and a new hand gesture g_{out}, which is conditioned on the goal gesture s.

[1]We denote the processed output image as I' to represent it is modified base on I.

In alignment, we first use $handDetect(\cdot)$ to produce the bounding box (bb_I) of the positive hand gesture in I and use $skeletonDetect(\cdot)$ to predict the bounding box ($bb_{\mathcal{K}}$) of the negative skeleton gesture in $\mathcal{K}$. Note that $handDetect(\cdot)$ can be any hand detection algorithm that produces a bounding box of the detected hand. In this work, we use MediaPipe (Lugaresi et al., 2019) as our hand detection model. We develop a simple $skeletonDetect(\cdot)$ algorithm to detect the left top and right bottom white pixels in a skeleton image as bounding box indices. With bb_I and $bb_{\mathcal{K}}$, we calculate the position of g_{in} and s. Note that in our alignment, we process input image I instead of $\mathcal{K}$, therefore, our calculated width and height offsets are for I. With width and height offsets, we align I by image processing techniques, i.e., padding and cropping, to reshape and resize I in order to make the hand position in I similar to the skeleton position in $\mathcal{K}$. We denote our aligned image as I' and pass it to GestureGAN for generation. Because there are paddings in I', the GestureGAN generated data has similar patterns as well as shown in Figure 2.

4 EXPERIMENT

In this section, we first introduce the datasets we used for training and test, the training details of two gesture detection models we considered, and the evaluation metrics. Then we provide ablation study to justify the benefit of our alignment algorithm and demonstrate the results of the proposed data augmentation pipeline.

4.1 Dataset

We train and evaluate our proposed framework on three benchmark datasets: Jester dataset (Materzynska et al., 2019), NTU Hand Digit (Ren et al., 2013), and Cre-

Algorithm 1: Align input image I and input skeleton $\mathcal{K}$ according to the positions of (original) hand gesture g_{in} and (goal) hand skeleton s.

Input : Image-skeleton pair $[I, \mathcal{K}]$.
Output : Aligned image-skeleton pair $[I', \mathcal{K}]$.

Hand and skeleton detection
  $bb_I = [x_I^0, y_I^0, x_I^1, y_I^1] \leftarrow handDetect(I)$
  $bb_{\mathcal{K}} = [x_{\mathcal{K}}^0, y_{\mathcal{K}}^0, x_{\mathcal{K}}^1, y_{\mathcal{K}}^1] \leftarrow skeletonDetect(\mathcal{K})$
end
Image-skeleton alignment
  Size of $I = [W_I, H_I]$
  Size of $\mathcal{K} = [W_{\mathcal{K}}, H_{\mathcal{K}}]$
  Position of $g_{in} = [x_I^{pos}, y_I^{pos}] \leftarrow [(x_I^0 + x_I^1) * W_I/2), (y_I^0 + y_I^1) * H_I/2)]$
  Position of $s = [x_{\mathcal{K}}^{pos}, y_{\mathcal{K}}^{pos}] \leftarrow [(x_{\mathcal{K}}^0 + x_{\mathcal{K}}^1) * W_{\mathcal{K}}/2), (y_{\mathcal{K}}^0 + y_{\mathcal{K}}^1) * H_{\mathcal{K}}/2)]$
  Width offset $\Delta w = x_I^{pos} - x_{\mathcal{K}}^{pos}$
  Height offset $\Delta h = y_I^{pos} - y_{\mathcal{K}}^{pos}$
  Process I for alignment:
  if $\Delta w > 0$ **then**
    $I' \leftarrow$ Pad I with Δw on the right hand side ▷ Pad
    $I' = I'[:, \Delta w :]$ ▷ Crop
  else
    $I' \leftarrow$ Pad I with Δw on the left hand side ▷ Pad
    $I' = I'[:, : \Delta w]$ ▷ Crop
  end
  if $\Delta h > 0$ **then**
    $I' \leftarrow$ Pad I' with Δh on the bottom ▷ Pad
    $I' = I'[\Delta h :, :]$ ▷ Crop
  else
    $I' \leftarrow$ Pad I' with Δh on the top ▷ Pad
    $I' = I'[: \Delta h, :]$ ▷ Crop
  end
end

ative Senz3D (Memo and Zanuttigh, 2018). Jester datasets is a diverse hand gesture video dataset which contains 27 classes. We select three positive gesture classes, i.e., hand raise, finger raise, and thumb up, and one negative gesture class, i.e., no gesture, for our real-world online meeting application scenarios. All positive gesture should trigger a reaction, e.g., hand raise and finger raise both trigger "Raise Hand" reaction, and thumb up triggers "Thumb Up" reaction. We split Jester dataset into Jester_train and Jester_test sets for training and test purposes. NTU Hand Digit dataset has diverse hand gestures including digits from zero to nine. We take all hand gestures (except digit 1, digit 5, and thumb up) as no gesture, and utilize digit 2, digit 3, digit 4, and rock roll gesture (with thumb, index finger, and little finger) as negative goal gestures for generation purposes. I.e., hand gestures for digit 2, 3, 4, and rock roll gestures are NTU_train, with which we utilize the skeleton images ($\mathcal{K}$) of NTU_train set in data augmentation. The rest gestures are used as test data (NTU_test). Later on, alignment algorithm shown in Algorithm 1 takes paired input $[I, \mathcal{K}]$ and produces $[I', \mathcal{K}]$ to train models. Creative Senz3D dataset contains ten different gestures, and we use this dataset as an independent test set to evaluate models and their augmented variants. Note that in all datasets, gestures that are not positive gestures are labeled as negative gesture.

4.2 Training Details

We consider two real-time models, 3D-CNN and MobileNetv2 (Sandler et al., 2018), for their real-time advantage to evaluate our augmentation pipeline in online meeting automatic reaction application. For MobileNetv2, we take the pre-trained version to further study the effectiveness of the propose augmentation pipeline. We train all models with Jester_train, which results in baselines without augmentation and we denote them as " Original Model" in later experimental result comparisons, and augmented dataset, which contains Jester_train and $\mathcal{D}^*$. We use the positive gesture finger raise as base in all generations.

For 3D-CNN, we use stochastic gradient descent (SGD) with an adaptive learning rate. We set the value of the initial learning rate to 10^{-2}, 10^{-5} for decay, and 0.9 for momentum. We train the model for 100 epochs with early stopping. For MobileNetv2, we take ImageNet (Russakovsky et al., 2015) pre-trained model from Tensorflow (tf,) and fine-tune it with the same setup used in 3D-CNN training.

Usage of GestureGAN. GestureGAN is an image-based model, so we utilize it to generate image frames as input to downstream models. The original design of GestureGAN makes it take a paired input, $[I_A, I_B, \mathcal{K}_A, \mathcal{K}_B]$, in both training and generation phase, where I_A is the input with a positive gesture, I_B is the output with a negative gesture, $\mathcal{K}_A$ is the input skeleton image of I_A, and $\mathcal{K}_B$ is the input skeleton image of I_B. However, in our application we do not need this paired input since we only require the generation of I_B based on input pair $[I_A, \mathcal{K}_B]$. Therefore, we replace I_A and $\mathcal{K}_A$ with blank placeholder images.

Evaluation Metrics. Accuracy, false positive rate, weighted precision, and weighted f1 score are used to evaluate the models quantitatively.

4.3 Ablation Study

In this section, we first study the effectiveness of our proposed alignment algorithm and inspect whether the performance of 3D-CNN is affected by the quality

of GAN generated data. To do this, we compare the generation quality of GestureGAN with and without the alignment algorithm. Next, we evaluate the benefit of different goal gestures by comparing "Peace sign", "OK sign", "4 fingers", and "Rock roll sign".

How Does Alignment Affect Model Performance? The proposed alignment algorithm is intended to improve the generation quality of GestureGAN and further help with hand detection model training. We will elaborate the benefit of alignment by comparing the performance of 3D-CNNs trained with $\mathcal{D}^*$ and $\mathcal{D}$.

We take "Peace sign" augmented 3D-CNNs and demonstrate the advantages of alignment by comparing 3D-CNNs trained with $\mathcal{D}^*$ (with alignment) and $\mathcal{D}$ (without alignment). From the results shown in Table 1, we see that model trained with $\mathcal{D}^*$ achieves higher results most of the time. Especially in NTU_test, it reduces FPR by 18.44% compared to model trained with $\mathcal{D}$. In the external Senz3D test set, model trained with $\mathcal{D}^*$ beats model trained with $\mathcal{D}$ in terms of all evaluation metrics we considered. These results elaborate the advantage of alignment in improving the hand detection model's performance. Because in real-world applications, we put final detection performance at a higher priority, alignment algorithm although adds some workload in the offline data pre-processing stage, we are convinced by its benefits brought to the online detection stage.

Different Goal Gesture Improves Model In Different Ways. We find in our experiments that different hand gesture has different effect in improving the models. In this section, we show the results empirically.

In this ablation study, we use four goal gestures separately in augmentation to enhance the 3D-CNN model, and we find different gesture improves models in different ways. Experimental results of accuracy, precision, false positive rate (FPR), and f1-score, of original and augmented 3D-CNN are shown in Table 2. We test our models with three datasets, where Jester_test and NTU_test provides general test results and an external Senz3D dataset helps evaluate the generalization ability of our augmented models. The results demonstrate that all four goal gestures improve the model, and "OK Sign" provides the best results compared to other gestures. It reduces FPR to 0% in NTU_test and improves FPR by 32.93% in the external Senz3D dataset, which indicates that the augmented model not only performs well with seen gestures, but also provides great performance when working with unseen gestures.

Table 1: Experimental results of "Peace sign" augmented 3D-CNN. $\mathcal{D}^*$ and $\mathcal{D}$ denote that the augmented models are trained with and without alignment. This comparison evaluates the contribution of the alignment algorithm and we can that the models benefit from alignment and achieves higher performance most of the time.

Test	Train	Accuracy	FPR	Precision	F1 Score
Jester_test	$\mathcal{D}^*$	**94.18%**	3.31%	**94.65%**	**94.27%**
	$\mathcal{D}$	93.84%	**3.01%**	94.48%	93.96%
NTU_test	$\mathcal{D}^*$	**55.41%**	**2.98%**	41.36%	**41.32%**
	$\mathcal{D}$	49.66%	21.42%	**63.24%**	43.67%
Senz3D	$\mathcal{D}^*$	**44.13%**	**60.06%**	**50.95%**	**42.38%**
	$\mathcal{D}$	38.26%	67.99%	45.47%	36.30%

4.4 Results on Pre-Trained Model

MobineNetv2 is known to be fast and effective in computer vision applications, in this work, we test our augmentation pipeline on MobileNetv2 and compare with 3D-CNN variations. Because MobineNetv2 is an image-based model, we take four consecutive frames as input and each frame is processed by a MobineNetv2 block independently. The results of pre-trained and fine-tuned MobileNetv2 are shown in Table 3. "OK sign", "4 fingers", and "Rock roll sign"-based augmentations all improve MobineNetv2 and achieve better performance in Jester_test, NTU_test, and Senz3D. However, comparing to 3D-CNN variants, MobineNetv2 variants although achieve better results under Jester_test, they all suffer from weak generalization ability, i.e., augmented MobineNetv2 variants have relative higher FPR in NTU_test and Senz3D. Overall, our proposed augmentation pipeline provides improvements in terms of all evaluation metrics under all datasets we tested with, while different goal gesture provides different improvements and both un-pre-trained model (3D-CNN) and pre-trained model (MobileNetv2) can benefit from our pipeline.

5 CONCLUSIONS

Serving real-time hand gesture recognition models faces several challenges in video conference software, such as real-time recognition, accurate prediction with low false alarm rate. Achieving these goal all together needs a serious amount of work in model developing and data collecting in a commercial setup. In this work, we develop a low-cost fast-to-market hand gesture recognition model to achieve real-time accurate prediction and propose a generation-based data augmentation pipeline to reduce false alarm rates without costing extra dollars in data collection. We demonstrate empirically that the models trained with augmented dataset achieve better results 95% of the time, and the false alarm rates reduce significantly. It is known that GAN-generated data sometimes does not look perfect via human inspection. In our experiments, we find

Table 2: Comparison of 3D-CNN and its variants trained with augmented datasets with four different negative gestures. Original model is 3D-CNN trained without data augmentation pipeline and it is the baseline we compare with. 3D-CNN variants trained with augmented data in general achieves better results, especially in NTU test set and Senz3D dataset. "OK sign" among all four negative gestures provides relatively better results in augmentation. 3D-CNN model trained with "OK sign" augmented data results in extremely low FPR in both NTU test set and Senz3D dataset.

	Jester Test Set				NTU Hand Digit Test Set				Creative Senz3D Dataset			
3D-CNN	Accuracy	FPR	Precision	F1 Score	Accuracy	FPR	Precision	F1 Score	Accuracy	FPR	Precision	F1 Score
Original Model	94.08%	**2.81%**	**94.70%**	94.20%	45.61%	29.76%	35.74%	40.00%	55.26%	39.33%	54.91%	53.32%
Peace Sign	**94.18%**	3.31%	94.65%	**94.27%**	55.41%	2.98%	**41.36%**	41.32%	44.13%	60.06%	50.95%	42.38%
OK Sign	94.02%	3.71%	94.45%	94.11%	**56.76%**	**0.00%**	32.21%	41.10%	**70.04%**	**6.40%**	**58.66%**	**62.50%**
4 Fingers	94.10%	4.01%	94.59%	94.21%	55.41%	2.98%	35.66%	41.11%	65.59%	15.55%	57.24%	60.89%
Rock Roll Sign	93.91%	2.91%	94.51%	94.03%	**56.76%**	0.60%	38.43%	**41.76%**	65.79%	16.46%	57.47%	61.25%

Table 3: Comparison of MobileNetv2 and its variants trained with augmented datasets with four different negative gestures. MobileNetv2 in these experiments are pre-trained with ImageNet and fine-tuned with our hand gesture dataset. From the results we can see that variants trained with augmented data achieves better results in all three test sets we considered. For pre-trained MobileNetv2, no single negative gesture emerges to be a clear winner, "OK sign", "4 fingers", and "Rock roll sign" all appeared to be beneficial.

	Jester Test Set				NTU Hand Digit Test Set				Creative Senz3D Dataset			
MobileNetv2	Accuracy	FPR	Precision	F1 Score	Accuracy	FPR	Precision	F1 Score	Accuracy	FPR	Precision	F1 Score
Original Model	95.80%	0.70%	96.24%	95.88%	43.58%	56.55%	52.14%	46.35%	45.95%	58.84%	51.83%	43.72%
Peace Sign	95.78%	1.60%	96.12%	95.84%	30.07%	74.40%	45.46%	32.51%	55.47%	40.85%	54.76%	53.08%
OK Sign	95.53%	1.50%	95.95%	95.61%	**48.99%**	**33.33%**	54.88%	**50.17%**	52.23%	41.16%	53.62%	50.97%
4 Fingers	**96.14%**	**0.30%**	**96.54%**	**96.21%**	34.80%	51.79%	43.27%	33.29%	38.87%	67.68%	48.33%	36.97%
Rock Roll Sign	96.05%	1.20%	96.43%	96.12%	31.76%	60.12%	**56.93%**	35.20%	**60.32%**	**34.15%**	**66.94%**	**58.87%**

that models can still benefit from those imperfect data and learn useful features to gain generalization ability. Therefore, our future work could start from studying which visual features are most significant and effective for hand gesture detection models to eliminate false alarms while maintaining high accuracy.

REFERENCES

Amazon web service. https://aws.amazon.com.

Google cloud platform. https://cloud.google.com.

Microsoft azure. https://azure.microsoft.com/en-us/.

Pre-trained mobilenetv2 from tensorflow. https://www.tensorflow.org/api_docs/python/tf/keras/applications/mobilenet_v2/MobileNetV2.

(2016). Segmental spatiotemporal cnns for fine-grained action segmentation. In Leibe, B., Matas, J., Sebe, N., and Welling, M., editors, *Computer Vision - 14th European Conference, ECCV 2016, Proceedings*, Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), pages 36–52. Springer Verlag.

Abavisani, M., Joze, H. R. V., and Patel, V. M. (2019). Improving the performance of unimodal dynamic hand-gesture recognition with multimodal training. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1165–1174.

Boukhayma, A., Bem, R. d., and Torr, P. H. (2019). 3d hand shape and pose from images in the wild. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10843–10852.

Brix, C., Bahar, P., and Ney, H. (2020). Successfully applying the stabilized lottery ticket hypothesis to the transformer architecture. *arXiv preprint arXiv:2005.03454*.

Chen, Y., Zhao, L., Peng, X., Yuan, J., and Metaxas, D. N. (2019). Construct dynamic graphs for hand gesture recognition via spatial-temporal attention. *arXiv preprint arXiv:1907.08871*.

Cisco Systems. Webex. https://www.webex.com.

Doosti, B., Naha, S., Mirbagheri, M., and Crandall, D. J. (2020). Hope-net: A graph-based model for hand-object pose estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6608–6617.

Ge, L., Ren, Z., Li, Y., Xue, Z., Wang, Y., Cai, J., and Yuan, J. (2019). 3d hand shape and pose estimation from a single rgb image. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10833–10842.

Goldin-Meadow, S. and Alibali, M. W. (2013). Gesture's role in speaking, learning, and creating language. *Annual review of psychology*, 64:257–283.

Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., and Bengio, Y. (2014). Generative adversarial nets. *Advances in neural information processing systems*, 27.

Google. Google meet. https://meet.google.com/.

Hinton, G., Vinyals, O., and Dean, J. (2015). Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*.

Iqbal, U., Molchanov, P., Gall, T. B. J., and Kautz, J. (2018). Hand pose estimation via latent 2.5 d heatmap regression. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 118–134.

Isola, P., Zhu, J.-Y., Zhou, T., and Efros, A. A. (2017). Image-to-image translation with conditional adversarial networks. In *Proceedings of the IEEE conference on*

computer vision and pattern recognition, pages 1125–1134.

Iverson, J. M. and Goldin-Meadow, S. (1998). Why people gesture when they speak. *Nature*, 396(6708):228–228.

Karpathy, A., Toderici, G., Shetty, S., Leung, T., Sukthankar, R., and Fei-Fei, L. (2014). Large-scale video classification with convolutional neural networks. In *2014 IEEE Conference on Computer Vision and Pattern Recognition*, pages 1725–1732.

Karras, T., Aila, T., Laine, S., and Lehtinen, J. (2017). Progressive growing of gans for improved quality, stability, and variation. *arXiv preprint arXiv:1710.10196*.

Kingma, D. P. and Welling, M. (2013). Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*.

Köpüklü, O., Köse, N., and Rigoll, G. (2018). Motion fused frames: Data level fusion strategy for hand gesture recognition. *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 2184–21848.

Liu, J., Liu, Y., Wang, Y., Prinet, V., Xiang, S., and Pan, C. (2020). Decoupled representation learning for skeleton-based gesture recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5751–5760.

Lugaresi, C., Tang, J., Nash, H., McClanahan, C., Uboweja, E., Hays, M., Zhang, F., Chang, C.-L., Yong, M. G., Lee, J., et al. (2019). Mediapipe: A framework for building perception pipelines. *arXiv preprint arXiv:1906.08172*.

Materzynska, J., Berger, G., Bax, I., and Memisevic, R. (2019). The jester dataset: A large-scale video dataset of human gestures. In *Proceedings of the IEEE/CVF International Conference on Computer Vision Workshops*, pages 0–0.

Memo, A. and Zanuttigh, P. (2018). Head-mounted gesture controlled interface for human-computer interaction. *Multimedia Tools and Applications*, 77(1):27–53.

Microsoft. Teams. https://teams.microsoft.com.

Min, Y., Zhang, Y., Chai, X., and Chen, X. (2020). An efficient pointlstm for point clouds based gesture recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5761–5770.

Mirza, M. and Osindero, S. (2014). Conditional generative adversarial nets. *arXiv preprint arXiv:1411.1784*.

Mueller, F., Bernard, F., Sotnychenko, O., Mehta, D., Sridhar, S., Casas, D., and Theobalt, C. (2018). Ganerated hands for real-time 3d hand tracking from monocular rgb. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 49–59.

Narayana, P., Beveridge, R., and Draper, B. A. (2018). Gesture recognition: Focus on the hands. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5235–5244.

Poon, G., Kwan, K. C., and Pang, W.-M. (2019). Occlusion-robust bimanual gesture recognition by fusing multi-views. *Multimedia Tools and Applications*, pages 1–20.

Pudipeddi, B., Mesmakhosroshahi, M., Xi, J., and Bharadwaj, S. (2020). Training large neural networks with constant memory using a new execution algorithm. *arXiv preprint arXiv:2002.05645*.

Radford, A., Metz, L., and Chintala, S. (2015). Unsupervised representation learning with deep convolutional generative adversarial networks. *arXiv preprint arXiv:1511.06434*.

Rajbhandari, S., Rasley, J., Ruwase, O., and He, Y. (2020). Zero: Memory optimizations toward training trillion parameter models. In *SC20: International Conference for High Performance Computing, Networking, Storage and Analysis*, pages 1–16. IEEE.

Ren, Z., Yuan, J., Meng, J., and Zhang, Z. (2013). Robust part-based hand gesture recognition using kinect sensor. *IEEE transactions on multimedia*, 15(5):1110–1120.

Rezende, D. J., Mohamed, S., and Wierstra, D. (2014). Stochastic backpropagation and approximate inference in deep generative models. In *International conference on machine learning*, pages 1278–1286. PMLR.

Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M., Berg, A. C., and Fei-Fei, L. (2015). ImageNet Large Scale Visual Recognition Challenge. *International Journal of Computer Vision (IJCV)*, 115(3):211–252.

Sandler, M., Howard, A., Zhu, M., Zhmoginov, A., and Chen, L.-C. (2018). Mobilenetv2: Inverted residuals and linear bottlenecks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4510–4520.

Simonyan, K. and Zisserman, A. (2014). Two-stream convolutional networks for action recognition in videos. In Ghahramani, Z., Welling, M., Cortes, C., Lawrence, N., and Weinberger, K. Q., editors, *Advances in Neural Information Processing Systems*, volume 27. Curran Associates, Inc.

Tang, H., Liu, H., and Sebe, N. (2020). Unified generative adversarial networks for controllable image-to-image translation. *IEEE Transactions on Image Processing*, 29:8916–8929.

Tang, H., Liu, H., Xiao, W., and Sebe, N. (2019). Fast and robust dynamic hand gesture recognition via key frames extraction and feature fusion. *Neurocomputing*, 331:424–433.

Tay, Y., Bahri, D., Yang, L., Metzler, D., and Juan, D.-C. (2020). Sparse sinkhorn attention. In *International Conference on Machine Learning*, pages 9438–9447. PMLR.

Wang, L., Qiao, Y., and Tang, X. (2015). Action recognition with trajectory-pooled deep-convolutional descriptors. In *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4305–4314.

Zafrir, O., Boudoukh, G., Izsak, P., and Wasserblat, M. (2019). Q8bert: Quantized 8bit bert. *arXiv preprint arXiv:1910.06188*.

Zaheer, M., Guruganesh, G., Dubey, K. A., Ainslie, J., Alberti, C., Ontanon, S., Pham, P., Ravula, A., Wang, Q., Yang, L., et al. (2020). Big bird: Transformers for longer sequences. In *NeurIPS*.

Zhang, B., Wang, L., Wang, Z., Qiao, Y., and Wang, H. (2016). Real-time action recognition with enhanced

motion vector cnns. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2718–2726.

Zhu, J.-Y., Park, T., Isola, P., and Efros, A. A. (2017). Unpaired image-to-image translation using cycle-consistent adversarial networks. In *Proceedings of the IEEE international conference on computer vision*, pages 2223–2232.

Zoom. Zoom video communications. https://zoom.us/.