Sign Language Interpretation

Computer Vision for gesture recognition

- Jayasudan Munsamy

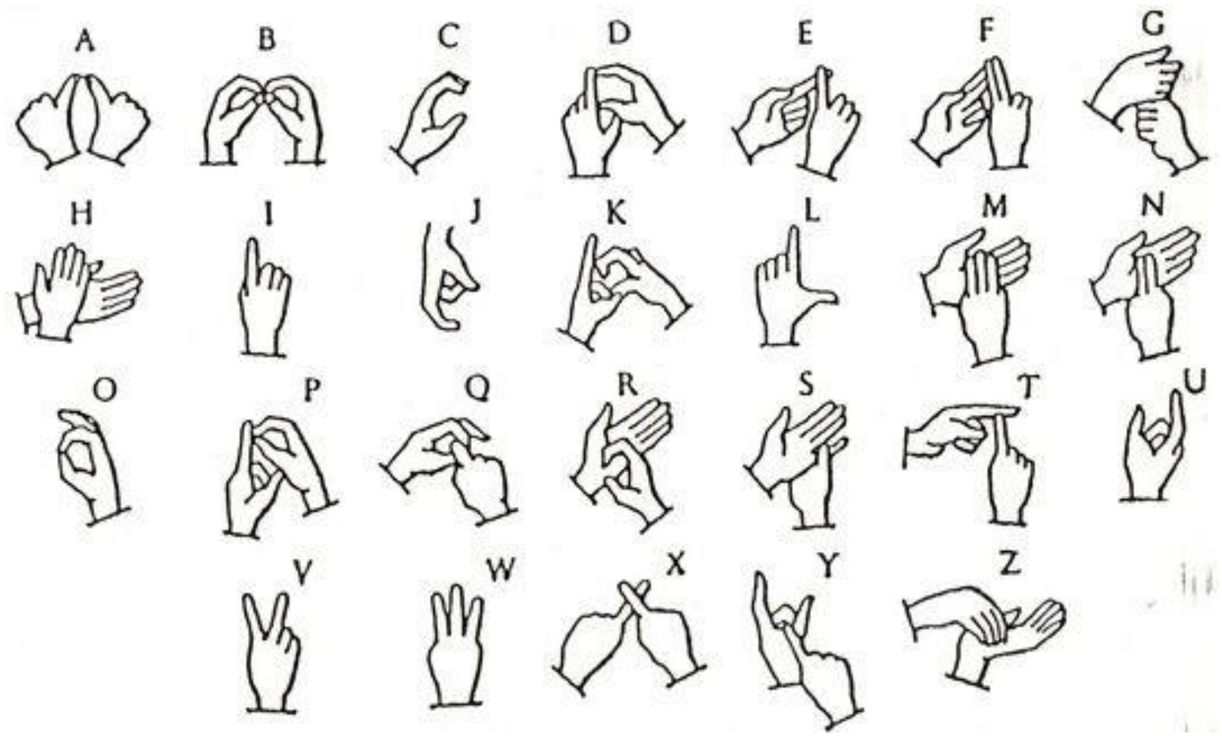
Founder & CEO of DeepVisionTech.AI

Contents

- About me
- Sign Language
- Challenges interpreting Sign Language
- Possible approaches
- Options for Image based recognition
- Options for Video based recognition
- Inference on smartphone
- Scope & recommendations
- Benchmarks
- Resources
- Demo videos
- About DeepVisionTech.AI

Sign Language

- Differences in sign languages followed by different countries – <u>single hand / double hand</u>
- Every state / city / village follow <u>one major sign language</u>, but with their own <u>dialects/variations</u>
- Has its own <u>grammar</u>

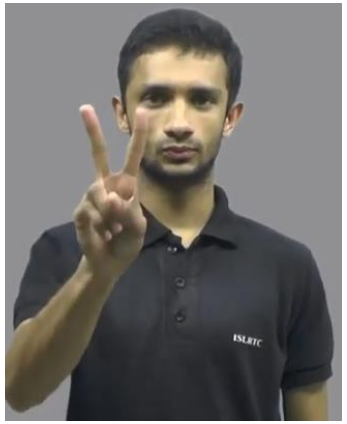


Challenges interpreting Sign Language

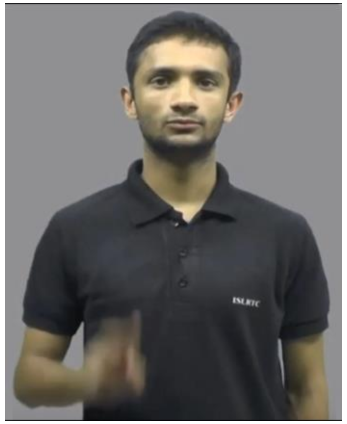
- Lack of labelled dataset
- Sign Language variations – number of hands, finger-spelling, word or phrase, various dialects/variations
- Interpreting grammar
- Facial expressions to consider
- Body language / movement to consider
- Variations in speed of sign'ing
- Spatial + Temporal data to consider – video sequence
- Huge vocabulary to cover
- Preprocessing complexity & time taken
- Choice of data augmentation
- Long model training duration
- Model size and inference duration
- Lack of baseline models and performance benchmarks

Possible approaches

Image



Video



1. **Image based classification** – use images for training & inference
    - Ok for sign language alpha-numerals, but not ok for words / phrases / sentences
    - Only finger-spelling based sign language interpretation is possible (demo)
    - Solid preprocessing techniques have to be defined and can be reused
    - Model can be pre-trained or custom classification / object detection model
    - Additional algorithms / logic to be implemented to form finger-spelled words

2. **Video based interpretation** – use video for training & inference
    - Good for alpha-numerals / words / phrases / sentences
    - Consider facial expression & body movements as well
    - Split video stream into sequence of frames at fixed intervals
    - Define number of frames to use & interval to pick frames – variations to be dealt with
    - Solid preprocessing techniques have to be defined
    - Additional algorithms / logic to be implemented to form phrases / sentences
    - Based on word / phrase interpreted, additional NLP model is required to form grammatically correct sentences

Options for Image based recognition

1. **SVM:** preprocess + feature descriptor HOG + HOG feature vector + SVM classification
2. **Transfer learning 1:** preprocess + pre-trained image classification model (VGG or Inception or so)
3. **Transfer learning 2:** preprocess + pre-trained object detection model (SSD or R-CNN)
4. **Custom Conv:** preprocess + extract features with a pre-trained model + custom ConvNet
5. **Pre-trained custom Conv:** preprocess + pre-trained custom ConvNet

Key points

- Similar to any image classification technique, but more focus to be given on dataset & preprocessing
- Classification models pre-trained on action detection can improve performance and needs less data
- Start with images with just hands in plain background then move on to images with person in plain background
- Decide the image dimensions carefully
- Images in dataset to consider aspects like multiple people with different appearances, left-handed person, etc.
- Data augmentations should be chosen carefully to take care of variations

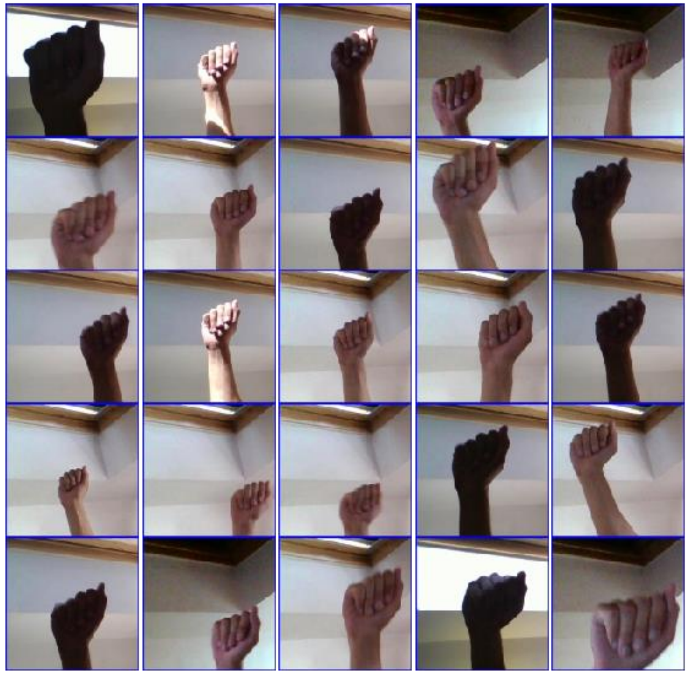
Image based recognition explained

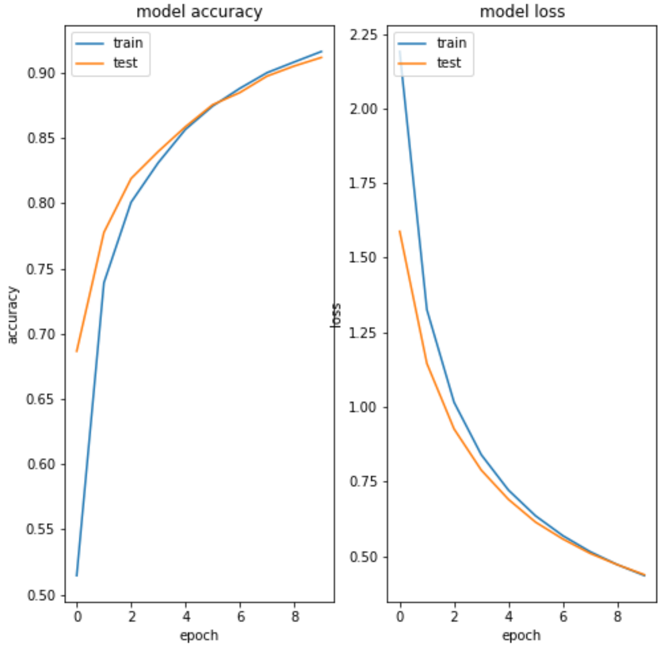
Transfer learning 1: Interpret sign language with Deep Learning (kaggle by paultimothymooney)

- Dataset: ASL A to Z, del, nothing, space, unknown : 30 classes, ~3k each, ~87k total
- Split data into training, validation & test – preferred is 70% + 15% + 15%
- Check the images to:
  - understand dataset – image size is 50x50, photos are of same angle but diff distance & position
  - understand if classes are balanced and fix incorrect labels
- Use transfer learning on VGG16
  - 3×3 conv layers stacked on top of each other in increasing depth. Reducing volume size is handled by max pooling. 2 fully-connected layers, each with 4096 nodes, followed by a softmax classifier
  - Freeze all layers except few near softmax & train the model with new dataset
  - Use Adam or RMSprop or SGD optimizer at a lower learning rate (0.0001)
- Define model checkpoints & train
- Plot metrics graphs
- Validation accuracy after just 10 epochs is 92% & validation loss is 0.44

NOTE: No pre-processing or data aug or feature extraction from images done…so, model may not be generalized enough and if we use different ‘test’ set of images, accuracy may be low

ASL dataset alphabet A





Options for Video based recognition

1. **2D ConvNet:** preprocess + extract features from video frames sequence + ConvNet for classification
2. **SVM or KNN:** preprocess + detect edges / extract features from video frames sequence + SVM or KNN for classification
3. **3D ConvNet** (arXiv)**:** preprocess + extract features from video frame sequence using ConvNet 3D with 3x3x3 kernels + SVM for classification
4. **RNN** (arXiv)**:** preprocess + extract features from video frame sequence using ConvNet + RNN
5. **RNN + soft Attention** (arXiv)**:** mainly suits video action description : preprocess + extract features from video frame sequence using ConvNet + features fed into deep RNN & attention mechanism for action prediction

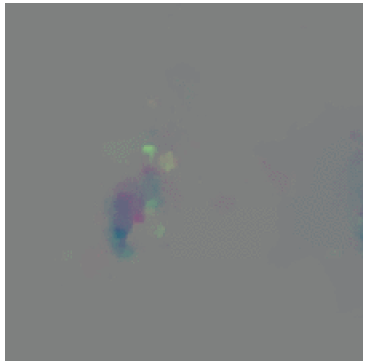
Two-streams mechanism (spatial & temporal input to two separate models)

1. **Inflated 3D Conv (I3D)** (arXiv)**:** preprocess + spatial stream with i3D (filters and pooling kernels of very deep image classification 2D Conv are expanded into 3D – NxN to NxNxN) + flow stream with i3D + average the predictions to get final prediction
2. **Hidden Two-Stream Conv Networks** (arXiv)**:** preprocess + 1 model generates optical flow (MotionNet) + temporal stream ConvNet takes in MotionNet's output to map motion info to action + weighted average of spatial stream ConvNet model's output and temporal stream output gives the prediction
3. **3D ConvNet & Attention** (arXiv)**:** mainly suits video action description (encoder CNN + decoder RNN + global temporal soft attention mechanism)
   - preprocess + extract local motion information using pre-trained Conv 3D + extract spatial features using pre-trained Conv 2D + concatenate 3D features with stacked 2D features + RNN for description generation

Two-streams mechanism explained

Sample video from UCF

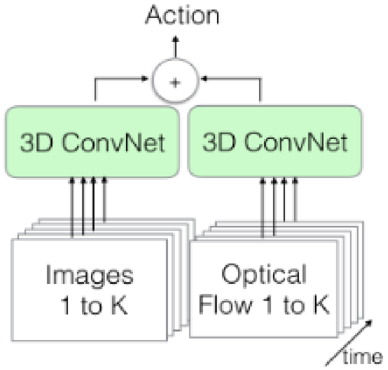
Optical flow of video



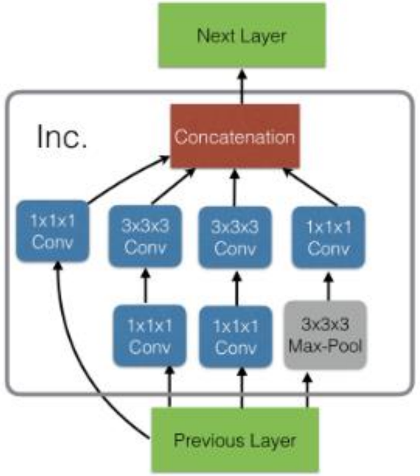
DeepMind's Inflated 3D Conv (I3D) + Kinetics 400 dataset (arXiv)

- Dataset: Kinetic Human Action video URLs dataset; 400 classes with ~400 videos per class, collected from realistic, challenging YouTube videos (link on how to download dataset)
- Split data into training, validation & test – preferred is 70% + 15% + 15%
- Check the videos to:
    - understand dataset – video dimensions, resolution, video length
    - understand if classes are balanced and fix incorrect labels (many YouTube links in dataset are broken)
- Preprocess: create optical flows for videos
- Use 3D model based on Inception-v1 for better performance, after pre-training on Kinetics dataset
    - Normal 3D ConvNet have high dimensionality & parameters making difficult to train and so we use shallow networks
    - Instead, deep networks like Inception, ResNet, etc. can be converted into spatio-temporal feature extractors
    - Add 3rd dimension to filters & kernels of Inception model (I3D)
    - First stream with I3D net is fed with RGB videos; second stream with I3D net is fed with flow videos
    - Average the predictions from both nets to get the final prediction
- Data augmentations – random cropping, random left-right flipping, resizing video to fit dimensions, random cropping a 224×224 patch, temporally start picking frames in sequence to ensure fixed number of frames, loop shorter videos as much time as possible

2-stream 3D ConvNet



2-stream 3D ConvNet



DeepMind's Kinetics 400 dataset + I3D

Detailed results comparison

NOTE: Model results will vary based on the preprocessing, pre-training and dataset used

Model	UCF-101	HMDB-51
Two-Stream [27]	88.0	59.4
IDT [33] (Improved Dense Trajectories)	86.4	61.7
Dynamic Image Networks + IDT [2]	89.1	65.2
TDD + IDT [34] (trajectory-pooled deep-conv descriptor)	91.5	65.9
Two-Stream Fusion + IDT [8]	93.5	69.2
Temporal Segment Networks [35]	94.2	69.4
ST-ResNet + IDT [7]	94.6	70.3
Deep Networks [15], Sports 1M pre-training	65.2	-
C3D one network [31], Sports 1M pre-training	82.3	-
C3D ensemble [31], Sports 1M pre-training	85.2	-
C3D ensemble + IDT [31], Sports 1M pre-training	90.1	-
RGB-I3D, Imagenet+Kinetics pre-training	95.6	74.8
Flow-I3D, Imagenet+Kinetics pre-training	96.7	77.1
Two-Stream I3D, Imagenet+Kinetics pre-training	**98.0**	80.7
RGB-I3D, Kinetics pre-training	95.1	74.3
Flow-I3D, Kinetics pre-training	96.5	77.3
Two-Stream I3D, Kinetics pre-training	97.8	**80.9**

Inference on smartphone

Refer **TensorFlow Lite** guide (link) or **Tensorflow JS**. Steps are for tflite.

Step1: Choose a trained TF model

Step2: Convert TF model with tflite converter tool

Step3: Optimize by quantizing (converting 32-bit floats to 8-bit int) – can be done while training or post training

Step4: Deploy model on device as .tflite file

Key Features

- Model optimization tools, including quantization, to reduce size and increase performance of models
- Tuned for devices, supports few core operators and small binary size
- Support Android and iOS devices, embedded Linux, and microcontrollers
- APIs for multiple languages including Java, Swift, Objective-C, C++, and Python
- High performance, with hardware acceleration on supported devices
- Pre-trained models for common machine learning tasks that can be customized

Scope & Recommendations for way forward

Phase 1 (2 weeks to complete)

Approach: Image based recognition (input to model is video frames)

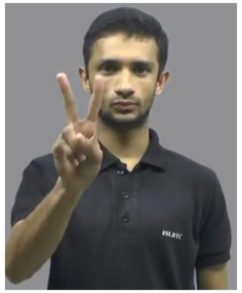
Dataset: Sign Language alphabets A to Z and / or numbers 1 to 10

Images & Model:

- Phase 1.1: just hands in plain background; pre-trained Conv
- Phase 1.2: person gesturing in plain background; pre-trained Conv / custom Conv
- Phase 1.3: implement algorithm to form words with interpreted alphabets
- Phase 1.4: deploy model on smartphone

Phase 1.1

Phase 1.2 (image)



Phase 2 (2 weeks to define possible solutions)

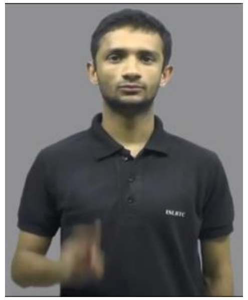
Approach: Video based recognition

Dataset: Sign Language gestures

Video & Model:

- Phase 2.1: person gesturing in plain background; one of the approaches mentioned earlier

Phase 2 (video)



Benchmarks

Rough guidelines for setting benchmarks.

For each sign language gesture (video):

Metrics	On Laptop	On Smartphone
Accuracy	**> 90%**	**> 90%**
Preprocessing	< 400ms	< 200ms
Inference	< 300ms	< 200ms
Overall	**< 700ms**	**< 400ms**

Resources (1/2)

Indian Sign Language: ISLRTC New Delhi (YouTube)

Image datasets:

- American Sign Language MNIST on kaggle – (link)
- EgoGesture Dataset – has both static & dynamic gestures (link)
- Hand Dataset (link)
- The 20BN-jester Dataset (link) – this is video dataset, but can create image dataset

Video datasets: (generic action & gestures)

- Bunch of action recognition related datasets (link)
- Kinetics 700 Dataset (link) – 650k videos from 700 action categories
- UCF101 - Action Recognition Dataset (link) – 13.3k videos from 101 action categories
- The 20BN-jester Dataset (link) – 148k videos from 27 action categories
- Kaggle's Hand gesture recognition Dataset – NIR images from Leap Motion sensor (link) – 2k videos from 10 action categories
- HMDB51 Dataset (link) – 7k clips from 51 action categories
- ASL Dataset (link by Boston University)

Resources (2/2)

Google's Hand tracking

Articles:

- Hands detection (Medium article by Victor Dibia)
- Gesture recognition using 20bn-jester dataset (Medium article by 20bn)
- Hand tracking with MediaPipe framework that used for building multimodal - video, audio, any time series data (Blog & arXiv paper by Google)
- DeepSign ISL (Blog by Akshay Bahadur)
- Search in arXiv
- Search in www.researchgate.net

Repos:

- DeepMind's Kinetics dataset + I3D (GitHub)
- Interpret sign language with Deep Learning (kaggle by paultimothymooney)
- Search for mentioned arXiv papers' code on Github
- Google's MediaPipe framework (GitHub)

Image based interpretation of sign language as finger-spelling

Sample demo

Video based interpretation of sign language

Sample demo

Visit: https://LetsTalkSign.org

Demo video: [LinkedIn post]

About us

DeepVisionTech.AI

Realize your AI vision with us

Mission

We aspire to build solutions that bring positive impact on Society, by leveraging Machine Learning and especially Computer Vision techniques. To fuel our growth, we enable business solutions evolve to changing customer expectations by embedding ‘intelligence’ in them.

Opportunities

- Part-time / fulltime interns or ML enthusiasts to work on ML solutions (CV & NLP) for social good
- Volunteers for sign language dataset creation
- Partnership & Collaboration opportunities

Events

- Watch-out for CV & NLP based hackathon coming soon

Connect

Email: Jayasudan@DeepVisionTech.AI

Phone: +91 97422 04284

LinkedIn: https://www.linkedin.com/in/jayasudan

Twitter: https://twitter.com/jayasudanm

Website: https://www.DeepVisionTech.AI

Solution Website: https://www.LetsTalkSign.org

Thank You