# SIGN LANGUAGE GESTURE DETECTION USINGDEEP LEARNING AND TRANSFER LEARNING

**Malla Asha, Sahithi Arekatla, Harsha Vardhan Gara, Dr. Gudikandhula Narasimha Rao**

Vignan's Institute of Information Technology (Autonomous), Visakhapatnam, Andhra Pradesh, India

ashamalla2k01@gmail.com, sahithiarekatla@gmail.com, harshavardhangara13350@gmail.com, gudikandhula@gmail.com

## Abstract

Physically challenged people like deaf and dumb face difficulty in communicating with others. They generally communicate using sign language gestures. Every normal person may not be aware of the sign language gestures, and it is challenging to learn the language. There have been various technological improvements, as well as much research, to aid the deaf and dumb. Deep learning along with computer vision can be used too to make an impact on this issue. In this project, a sign detector is created that identifies the gestures so that normal people can also understand what they are trying to convey. Further, this project uses voice assistance and text translation into multiple languages for better communication.

**Keywords**: Deep Learning (DL), Computer Vision (CV), Convolutional Neural Network (CNN), Residual Neural Network (ResNet50), TensorFlow Object Detection, TensorFlow JS

## I. Introduction

Sign language is a type of communication used by those who have difficulty in speaking or hearing. Disabled People make use of signs or gestures as a non-verbal communication tool to express their thoughts and emotions to others. However, because these ordinary people have a hard time understanding their expressions, experienced sign language experts are required. There is an upsurge in demand for such services in recent years. Some other kinds of services, such as video remote human interpreters requiring a high bandwidth internet connection, is established, providing an easy to use sign gesture interpreter service that may be utilized and benefited [1].

In order to address this problem, several algorithms like Convolutional Neural Network (CNN), Residual Neural Network (ResNet50), and Tensorflow Object Detection (SSD MobNet) were used to choose the best algorithm which gives accurate results for this problem.

Initially, images were collected from the web, and also a form was created and the images from our university students were collected to get more variations and make the model efficient. The analysis of the paper is organized as follows: Section II gives a summary of the performed related work; Section III briefs about the existing system and Section IV about the proposed one. Section V gives a detailed description of the insights of the dataset. Section VI overviews the various algorithms used and Section VII discusses the results obtained. At last, Section VIII expresses the conclusion and possible developments.

## II. Literature Survey

Several kinds of research have been done focusing to make communication easy for deaf and dumb people. Different algorithms and data processing techniques were used and accomplished unique accuracies and results.

J. Rekha, J.Bhattacharya, and S. Majumder [2] proposed two novel approaches for hand sign recognition that will detect sign language gestures in real-time environment. In their research, Knearest Neighbor and Support Vector Machine (SVM) are used for hybrid classification of single signed letters. Additionally, they proposed finger spelled word recognition using Hidden Markov Model (HMM) for a lexicon-based approach. The advantages of SURF and Hu Moment Invariant methods were combined and used as a combined feature set to achieve a better detection rate. The obtained the results as follows:

SIFT- 68%, Hu moment invariant- 58%, SURF-84.6%, SURF Moment-86.2% and SURF Momentderived features- 88%.

A Research Gap on Automatic Indian Sign Language Recognition based on Hand Gesture Datasets and Methodologies [3] proposed by Kruti J Dangarwala, Dr. Dilendra Hiran is based on recognizing gestures under dynamic parameters such as lighting conditions, multiple hands as background, left handed person, right handed person, size of finger, etc. are thoughtprovoking field in automating Indian sign language recognition. In this paper, Support Vector Machine, K-nearest neighborhood, Hidden Markov model, Fuzzy Interface System, Random Field Classifier, har cascade classifier for classifying the images and used HOG method, SIFT method, DWT approach, PCA approach, fusion approach for feature extraction. This research achieves a recognition rate of 97.5% through the Support Vector Machine classifier.

Dr. Arun Mitra, Nakul Nagpal, and Dr. Pankaj Agarwal [4] proposed a system to aid communication of deaf and dumb people communication with ordinary people using Indian sign language (ISL) where dynamic gestures will be automatically converted into a text message in real time. Finally, after testing is done the system will be implemented on the android platform and will be available as an application for smartphones and tablet pc.

## III. Existing System

Previously, people used to communicate with hearing and speech challenged people by learning their language and there was no such technical implementation to the problem at that time. Definitely, a translator should be there who converts the sign language to understandable form. Currently, there are few online live translators available which require high bandwidth internet and possess significant limitations. Presently available systems use machine learning algorithms which are best suitable for datasets with few images and fewer data processing tasks.

*Disadvantages :*

The following are some of the disadvantages of the existing system:

- Requires high bandwidth internet speed.
- A translator should be available with the person at all times.
- Machine learning algorithms are suitable for only datasets with few classes of images.

## IV. Proposed Scheme

In this system, deep learning computer vision technology has been used to automatically detect sign language. Deep learning algorithms such as CNN and ResNet50 have been used in which the model is trained with a few classes of sign language images and tested the model with new images. Tensorflow JS object detection is used which trains the model instantly in the browser and detects the gestures. The detection is also done using Tensorflow Object Detection (Transfer Learning). In this technique, the gesture images are labelled with signature names using LabelImg. Then the SSD MobileNet(deep learning) model is trained using transfer learning and detects sign language gestures in real-time using OpenCV.

*Advantages :*

- The user will be able to communicate in different languages.
- Internet access is not required for using this application.
- A translator is not required at all times.

### *A. About SSD MobNet*

In computer vision, creating accurate machine learning models which are capable of localising as well as detecting many objects in a single image is a major challege. The TensorFlow Custom Object Detection API is an open-source framework based on TensorFlow which enables building, training and deployment of object detection models very easily.

### *B. Methodology For Proposed System*

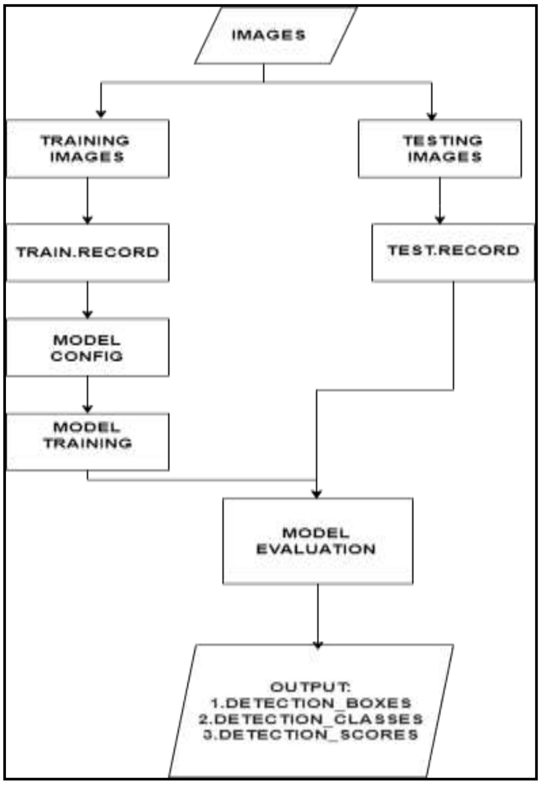


Fig. 1. Methodology for Proposed System

## V. Dataset Description

After exploring various data sources like Kaggle, AmazonS3, GitHub, and Google datasets there is no appropriate dataset suitable for the problem statement. The ISL-CSLTR dataset (Indian Sign Language Dataset for Continuous Sign Language Translation and Recognition) [4] is one of the datasets which is explored. This is a large dataset of size 8.5 GB which consists of the vocabulary of 700 fully annotated videos and 18,863 sentence-level image frames as well as 1,036-word level images for over 100 Spoken language Sentences which are performed by 7 different Signers which was one of the better choices of datasets that have been explored. However, this couldn"t be taken forward for this problem statement because of the following reasons:

- Although there are many classes of images of the signs, there is a very less number of images in each class which is not even sufficient for training a basic deep learning model.
- The picture quality of the images is not good.
- The size of the dataset is very large.

Considering the above issues, a dataset has been created with few classes of sign gestures by collecting images from the internet search and also collected images of the signs by capturing lively from different people. The dataset consisted of 8 classes of sign gesture images of which for each class there are 50 images. This dataset is used for the training of CNN and ResNet50 algorithmic models. Further, a custom dataset has been created for training the Tensorflow Object Detection (SSD MobileNet) algorithm. The dataset was created by capturing images instantly using the webcam by using OpenCV.

## VI. Algorithms

### *A. Convolutional Neural Network ( CNN )*

First of all, the images have been split into the train, test, and validation sets, and the images are rescaled to 255x255 pixels. To build the layers for the CNN model TensorFlow Keras Sequential Model is used. The layers include 3 convolutional, 3 pooling, 2 dense and 1 flatten layers. The model has been optimised using the RMSprop optimizer which is a gradientbased optimization technique used in training neural networks and using categorical cross entropy to calculate the loss and accuracy as the metric. For training the model, 30 epochs with 3 steps in each epoch, and a validation dataset has been used for validating the data. The test dataset is used for predicting the model's output.

### *B. Residual Neural Network (ResNet50)*

ResNet50 has 48 convolution layers, one max pooling layer, and an average pooling layer which has 3.8 x 10^9 floating point operations . It's the most popular ResNet model with 50 layers.
Fig. 2. Describes about the architecture of ResNet50 algorithm.

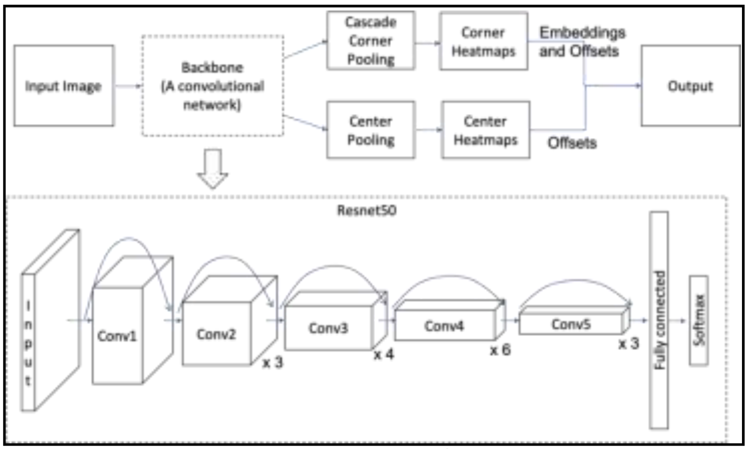


Fig. 2. Architecture of ResNet50

### *C. Transfer Learning using SSD MobNet ( Custom Model )*

SSD MobNet means Single Shot MultiBox MobileNetwork.
SSD Mobilenet V2 is an object detection model with FPN-lite (Feature Pyramid Network-lite) feature extractor, shared box predictor, and focal loss, which is trained on COCO 2017 dataset with training set images scaled to 320x320 initialized from Imagenet classification checkpoint. Model is created using the TensorFlow Custom Object Detection API. The following explanation gives a better understanding of SSD:

- Single Shot: It indicates that tasks of classification as well as object localisation are finished in a single forward pass of the network.
- MultiBox: Szegedy et al created this which is a bounding box regression technique. It is a method for quick class-agnostic bounding box coordinate recommendations, inspired SSD's bounding box regression algorithm.Interestingly, in the work done on MultiBox, an Inception-style convolutional network is used.
- Detector: The network is an detects the object and can also classify the objects it finds.

Fig. 3. describes about that architecture of SSD MobNet

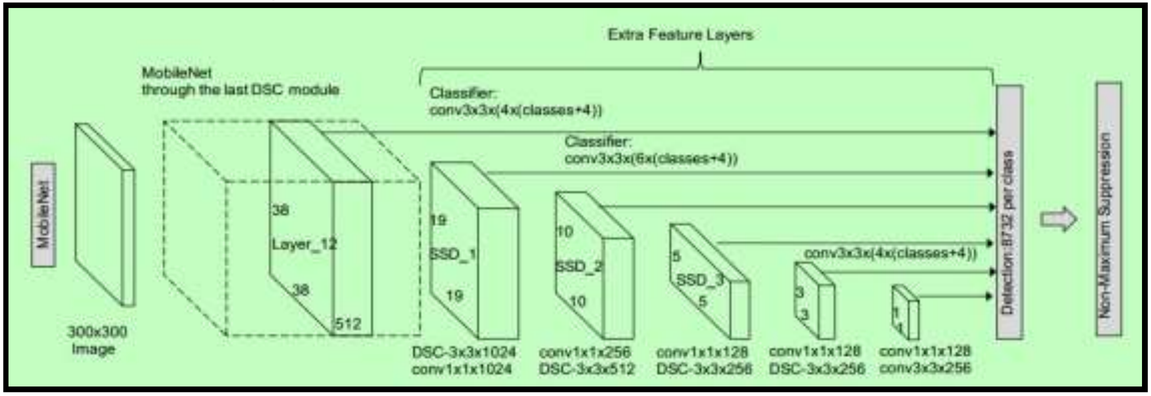


Fig. 3. SSD MobNet Architecture

Training a custom TensorFlow transfer learning model needs TensorFlow and Object Detection API. It involves preparing dataset, annotating dataset, partitioning & creating TFRecords. Firstly, install the latest version of Python. Download and install the Anaconda packages for the operating system supported from here [5]. Make sure to install TensorFlow 2. Then go to url [6] and download Visual

Studio C ++ 2015. This is required for Tensorflow to compile.Download and install Cuda [7] and Cudnn [8]. Cuda must be installed first, followed by Cudnn. Cudnn files must be copied into their corresponding folders within the Cuda directory once it has been installed [9]. Then install Protocol buffer [10]. For Windows OS, download the repository and then add the bin path to the PATH environment variable. Then, using the pip command, install python packages TensorFlow and openCV. Install the object detection API [12]. Prepare the dataset using opencv and capture the dataset. Annotation is done using LabelImage package which generates xml files for each image. After annotation, the data is parted into train and test data (Usually in 9:1 respective ratio) which is then converted into label map that maps the labels to integer or numerical values. Usually a label map file has .pbtxt extension. Annotations are then converted to TFRecord format which converts xml files to record format. Get the most recent pre-trained network for the model by simply clicking on the relevant model's name in the TensorFlow2 Object Detection Model Zoo's table. Some *.tar.gz file should be downloaded by clicking on the model's name. After downloading the *.tar.gz file, open it using unzipping software. After that, unzip the *.tar folder and extract its contents into workspace/pre-trained-models. Create a directory for the training job in the workspace/models create a new directory named my_ssd_mobnet and copy the configuration file in the path location workspace/pretrainedmodels/ssd_mobilenet_v2_fpnlite_320x320_coco17_tpu-8/pipeline.config file into the newly created directory. Now copy the TensorFlow/models/research/object_detection/model_main_tf2.py file and then paste it into the workspace folder before starting training the model. This script will be required to start training the model. Run necessary commands to initiate a training the model.

*Monitor Training Progress with TensorBoard*

Tensoboard platform a feature of TEnsorFlow, has been used to repeatedly monitor as well as visualize a variety of training & evaluation metrics. Fig. 4 gives the graphical representation of the classification and localization loss, Fig. 5 describes the charts of regularization and total loss, Fig. 6 gives chart of learning rate of the model.

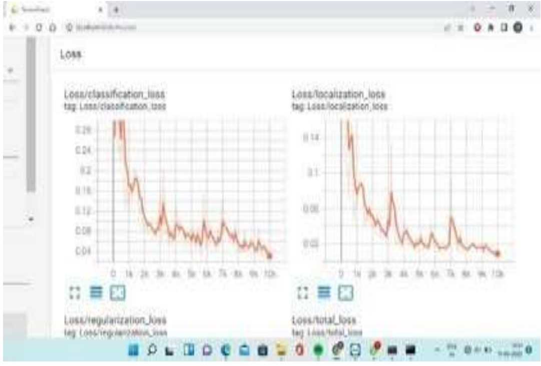


Fig. 4. Charts for Classification Loss and Localization Loss

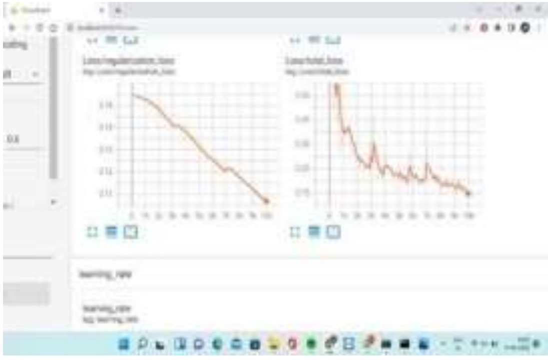


Fig. 5. Charts of Regularization loss and Total loss

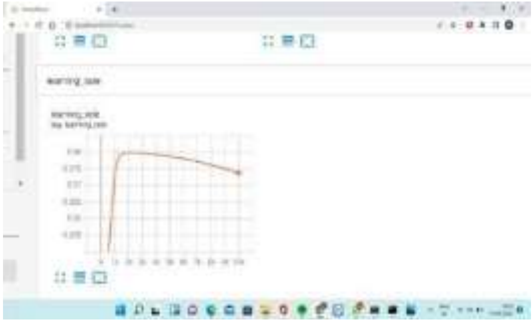
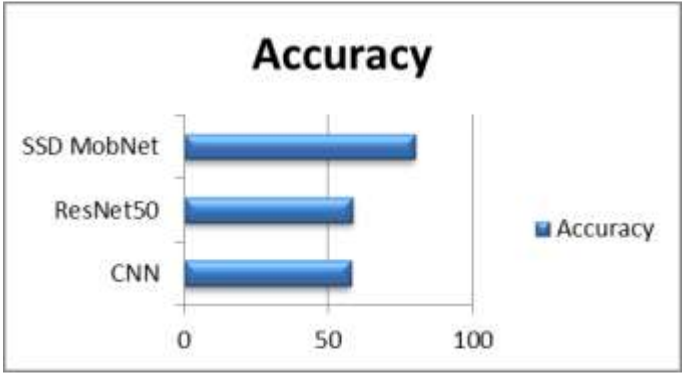


Fig. 6. Chart of Learning Rate

## VII. Results and Analysis

In this analysis, the results from three distinct calculations CNN, ResNet50, Transfer Learning using SSD MobNet which are regulated calculations are obtained. It is observed that CNN and ResNet gave accuracies between 50%-60%. SSD MobNet algorithm gave an accuracy of 80% which is highly proficient in deep learning and the accuracy could be increased by adding more images and using Cuda and GPU which would speed up the training process of the model.

CHART 1 Accuracies of different algorithms



From the result analysis, it is observed that the Transfer Learning using SSD MobNet algorithm gets a high accuracy of 80.056% which is much better than any other models. By using this model, the sign language gestures can be detected efficiently. It is an invention that can help the impaired community(deaf & dumb) to progress.

The Fig. 7. shows the prediction of the sign gesture “Yes” with a confidence level of 93%.

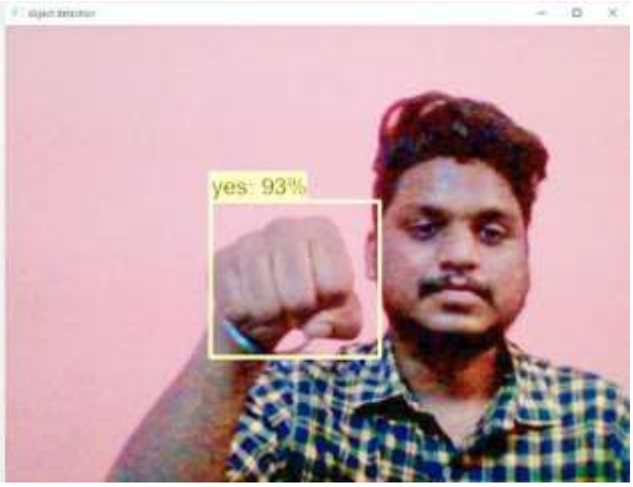


Fig. 7. Yes sign prediction

## VIII. Conclusion and Future Work

In this paper, the larger dataset with more classes can be used in the future. For this system, high processing power is needed to process the data that requires high-end hardware support which may be used in further future work. High-end hardware can accelerate the training and is also used for training complex algorithms. GPUs are the main component in the hardware requirement that is not used in this system and the usage of GPUs could give the best results with less time. This paper proposes the sign language gesture detection in which the meaning of the gesture is converted to the text where the image of the gesture is taken as input. For detecting gestures of different signs algorithms like CNN, ResNet, and SSD MobNet (Custom Tensorflow Transfer Learning Model) have been used. From the result, the proposed system best works with Custom SSD MobNet Model (80.056 %). The deaf and dumb people will be able to communicate with others effectively and also sign language need not be learned by everyone. High-speed internet is not required and communication can be simply done through a webcam facility.

## IX. Acknowledgement

This paper is a part of B. Tech final year major project. The completion of this undertaking could not have been possible without the assistance of our college – Vignan's institute of Information Technology, Duvvada, Visakhapatnam, Andhra Pradesh 530049, India. Their contributions and guidance are gratefully acknowledged. Their support and contribution ensured the completion of the project. We are highly indebted to our Principal, our Training and Placement Dean, and our training and placement cell, who have granted us with necessary data and made it possible for us to work on it. We express our gratitude to our project review panel for gracing us with their advice. We are highly grateful to our guide Dr. Gudikandhula Narasimha Rao, who has put us in the right path and has guided us through the project from the start to the end with his abundant experience and has ensured to the completion of the paper.

## References

[1] Dr. R.S. Sabeenian, S. Sai Bharatwaj, M. Mohamed Aadhil, Sign Laguage Recognition using Deep learning and Computer Vision, Jour of Adv Research in Dynamical & Control Systems, Vol. 12, 05-Special Issue, 2020.

[2] J.Rekha, J.Battacharya, S. Majumder, "Hand Gesture Detection for Sign Language: A New Hybrid Approach".

[3] Kruti J Dangarwala, Dr. Dilendra Hiran, " A Research Gap on Automatic Indian Sign Language Recognition based on Hand Gesture Datasets and Methodologies", International Journal of Computer Engineering and Applications, Volume XII, Issue III, March 18, www.ijcea.com ISSN 2321-3469.

[4] Dr. Arun Mitra, Nakul Nagpal, and Dr. Pankaj Agarwal, "Design Issue and Proposed Implementation of Communication Aid for Deaf & Dumb People", International Journal on Recent and Innovation Trends in Computing and Communication ISSN: 2321-8169 Volume: 3 Issue: 5 147 – 149.

[5] Indian Sign Language Dataset for Continuous Sign Language Translation and Recognition, https://data.mendeley.com/datasets/kcmpdxky7p/1

[6] Anaconda Versions, https://repo.anaconda.com/archive/

[7] Visual Studio C++ Build Tools, https://go.microsoft.com/fwlink/?LinkId=691126

[8] CUDA Toolkit10.1 archive, https://developer.nvidia.com/cuda-10.1-download-archivebase

[9] Cudnn: 7.6.5 , https://developer.nvidia.com/rdp/cudnn-download

[10] https://towardsdatascience.com/installing-tensorflow-with-cuda-cudnn-and-gpu-support-on-windows-10-60693e46e781

[11] Gudikandhula Narasimha Rao, et al, "Fire detection in Kambalakonda Reserved Forest, Visakhapatnam, Andhra Pradesh, India: An Internet of Things Approach", Journal of Materials Today: Proceedings, Elsevier, Volume 5, Issue 1, Pages: 1162–1168, 2018, ISSN: 2214-7853.

[12] Rao, G. Narasimha, R. Ramesh, and D. Rajesh. "D. Chandra sekhar." An Automated Advanced Clustering Algorithm For Text Classification"." International Journal of Computer Science and Technology 3.2-4 (2012).

[13] Rao, Gudikandhula Narasimha, and P. Jagdeeswar Rao. "A Clustering Analysis for Heart Failure Alert System Using RFID and GPS." ICT and Critical Infrastructure: Proceedings of the 48th Annual Convention of Computer Society of India-Vol I. Springer, Cham, 2014.

[14] Rao, Gudikandhula Narasimha, et al. "Geo Spatial Study on Fire Risk Assessment in Kambalakonda Reserved Forest, Visakhapatnam, India: A Clustering Approach." Proceedings of International Conference on Remote Sensing for Disaster Management. Springer, Cham, 2019.

[15] Rao, G.N., Rao, P.J., Duvvuru, R.: A drone remote sensing for virtual reality simulation system for forest fires: semantic neural network approach. In: IOP Conference Series: Materials Science and Engineering. vol. 149. no. 1. IOP Publishing, 2016

[16] Rao, G.N., et al.: An enhanced real-time forest fire assessment algorithm based on video by using texture analysis. Perspect. Sci. 8, 618–620 (2016)

[17] Protocol Buffers , https://github.com/protocolbuffers/protobuf/releases

[18] Custom Tensorflow SSD MobNet v2 fplite 320x320 Model in Tensorflow Hub ,https://tfhub.dev/tensorflow/ssd_mobilenet_v2/fpnlite_320x320/1

[19] Tensorflow Model Garden , https://github.com/tensorflow/model