**Title** : Indian Sign Language Translator Using Machine Learning

**Author** : Manasvi Srivastava

**Institution** : Banasthali Vidyapeeth

**Abstract**

Sign language is the efficacious medium that connects silent people and the world, and there are many significant existing sign languages across the globe. Communication is defined as the act of sharing or exchanging information, ideas or feelings. To establish communication between two people, both of them are required to have knowledge and understanding of a common language. But in the case of deaf and dumb people, the means of communication are different. Deaf is the inability to hear and dumb is the inability to speak. They communicate using sign language among themselves and with normal people but normal people do not take seriously the importance of sign language. Not everyone possesses the knowledge and understanding of sign language which makes communication difficult between a normal person and a deaf and dumb person. A lot of research is done to compromise the borders of difficulty in the communication between silent and normal person and most of them are based on ASL (American Sign Language). The aspiration for bridging the gap between silent people and normal people in terms of communication using ISL (Indian Sign Language) led to unfold this project. The existing Indian Sing Language Recognition systems are designed using machine learning algorithms with single and double-handed gestures but they are not real-time. In this paper, we propose a method to create an Indian Sign Language dataset using a webcam and then using transfer learning, train a TensorFlow model to create a real-time Sign Language Recognition system. The system achieves a good level of accuracy even with a limited size dataset.

**Keywords :**  Sign Language Recognition (SLR), Computer Vision, Machine Learning, Indian Sign Language.

1. **Introduction**

A sign is a hand motion or both hand movements combined with a facial expression that communicates a specific message. There are approximately 70 million silent people across the world and most of them are illiterate and communicate using signs. The nonverbal communication used by silent people is called sign language, and it is comprised of various gestures, each gesture has its meaning. These gestures are used to express their thoughts and feelings. There is a wide range of sign languages each having their own importance and significance. The Globally used sign language is American Sign Language (ASL). ASL includes the use of both hands to perform the sign language whereas In Indian sign language both hands along with facial expressions are used to express a gesture. These gestures are to be understood by vision. It can be considered successful only when whatever message the speaker is trying to convey is received and understood by the listener.

It can be divided into different categories as follows [1]: formal and informal communication, oral (face-to-face and distance) and written communication, non-verbal, grapevine, feedback, and visual communication, and the active listening. formal communication (official communication) is steered through the channels that are pre-determined. The unofficial or grapevine communication is the spontaneous communication between individuals in one’s profession that does not have any formal protocol or structure. The oral communication (face-to-face and distance) is the communication in which words are exchanged between people who are present in front or at a distance (with the help of technology including voice and video calls, webinars, etc.). The written communication is the communication in which letters, emails, notices, or any other written form is used for communicating. The non-verbal communication is the communication that uses gestures, facial expressions, body language, etc. The feedback communication happens when a person gives feedback on some product or service provided by an individual or a company. The visual communication occurs when a person gets information from a visual source like televisions, social networking, or any other source. Active listening is when a person listens to and understands what the other individual is trying to convey so that the communication becomes more meaningful and effective [1]. Non-verbal communication helps deaf and dumb people to communicate amongst themselves and with others. Deaf is a disability that impairs a person's hearing ability and makes them incapable to hear while dumb is a disability that impairs the speaking ability and makes them incapable to speak. Not being able to speak or listen makes it difficult to establish communication with others. This is where sign languages come into the role, it enables a person to communicate without words. But a problem still exists, not many people possess the knowledge of sign language. Deaf and dumb may be able to communicate amongst themselves using sign languages but it is still difficult for them to communicate with people having normal hearing and vice-versa due to the lack of knowledge of sign languages. This issue can be resolved by the use of a technology-driven solution. By using such a solution, one can easily translate the gestures of sign language into the commonly spoken language, English. A lot of research has been done in this field and there is still a need for further research. For gesture translation, data gloves, motion capturing systems, or sensors have been used [2]. Vision-based SLR systems have also been developed previously [3]. The existing Indian Sign Language Recognition system was developed using machine learning algorithms with MATLAB [4]. Authors have worked on single-handed and double-handed gestures. They used two algorithms to train their system, K Nearest Neighbours Algorithm and Back Propagation Algorithm. Their system achieved 93-96% accuracy. Though being highly accurate, it is not a real-time SLR system. The objective of this paper is to develop a real-time SLR system using TensorFlow, MediaPipe, OpenCV and train it using a dataset that will be created using a webcam.

The rest of this paper after the introduction is organized as follows.

**Section 2** presents the related work on the SLR system.

**Section 3** describes the data acquisition and generation.

**Section 4** focuses on the methodology of the developed system.

**Section 5** presents the experimental evaluation of the system.

And finally, **Section 6** concludes the paper with future work.

1. **Related Works**

This literature survey states various existing systems on the translation of the sign to text and vice versa using various methods. Sign languages are a highly organized set of hand gestures that carry specific meanings and are predominantly used by individuals with hearing impairments to facilitate daily communication.[3] These visual languages utilize the intricate movements of the hands, face, and body to express thoughts and ideas. Remarkably, there are over 300 different sign languages used around the world, each with its unique structure and vocabulary.[5] However, despite the wide variety of sign languages, the percentage of the global population fluent in any of them remains relatively low. This creates significant communication challenges for specially-abled individuals, as they may struggle to interact freely with those who do not understand sign language.

Sign Language Recognition (SLR) systems provide a promising solution to this issue by enabling communication in sign language without the need for prior knowledge. These systems are designed to recognize and interpret hand gestures, translating them into commonly spoken languages such as English. Through SLR, specially-abled individuals can communicate more effectively with the broader population, bridging the gap and fostering inclusivity in everyday interactions. Sign Language Recognition (SLR) is a broad and complex field of research that, while already extensively studied, still presents numerous challenges that need to be addressed. Machine learning techniques empower electronic systems to make decisions based on accumulated experience, which is essentially data. To implement classification algorithms, two primary datasets are required: a training dataset and a testing dataset. The training dataset provides experiences to the classifier, and the model's performance is evaluated using the testing dataset [6]. Numerous researchers have developed efficient methods for data acquisition and classification [3][7].

Previous work in SLR can be categorized based on the data acquisition method into two main approaches: direct measurement methods and vision-based approaches [3]. Direct measurement methods utilize tools such as motion data gloves, motion capturing systems, or sensors. These methods are highly effective as the motion data collected allows for precise tracking of fingers, hands, and other body parts, contributing to the development of robust SLR methodologies.

On the other hand, vision-based SLR approaches rely on extracting discriminative spatial and temporal features from RGB images. These methods typically start by attempting to track and isolate hand regions before classifying them into specific gestures [3]. Hand detection in vision-based approaches is often achieved through semantic segmentation and skin color detection, as skin color tends to be easily distinguishable [8][9]. However, challenges arise because other body parts like the face and arms can sometimes be mistakenly identified as hands. To mitigate this, recent hand detection methods also incorporate face detection and subtraction, as well as background subtraction, to accurately identify only the moving parts within a scene [10][11].

To achieve precise and robust hand tracking, particularly in scenarios involving obstructions, researchers have employed filtering techniques such as Kalman and particle filters [10][12]. These techniques help in maintaining the accuracy and robustness of hand tracking, which is crucial for effective SLR systems.

For data acquisition, whether utilizing direct measurement methods or vision-based approaches, a variety of devices are necessary. The primary device used in the input process of an SLR system is a camera, which captures the visual data needed for gesture recognition [13]. Cameras are essential for vision-based approaches as they provide the raw RGB images from which discriminative spatial and temporal features can be extracted. In addition to standard cameras, advanced devices such as the Microsoft Kinect are also commonly used. The Kinect is particularly advantageous because it offers both a color video stream and a depth video stream simultaneously, with the depth data playing a crucial role in background segmentation.

Beyond traditional cameras and the Kinect, other devices are employed to capture the necessary data for SLR systems. For instance, accelerometers and sensory gloves are widely used in direct measurement methods. These devices can capture the precise movements of the hands and fingers, providing highly accurate motion data that is invaluable for developing robust SLR methodologies. Accelerometers measure the acceleration forces on the hands, while sensory gloves capture the flex and extension of the fingers, offering a detailed representation of hand gestures.

Another advanced system used for data acquisition in SLR is the Leap Motion Controller (LMC). The LMC is designed to track hand and finger movements with high precision, using infrared sensors to create a 3D representation of the user's hands. This allows for accurate detection and tracking of hand gestures in real-time, making it an excellent tool for SLR applications [14][15]. Approximately, it can operate around 200 frames per second and can detect and track the hands, fingers, and objects that look alike fingers. Most of the researchers collect their training dataset by recording it from their signer as finding a sign language dataset is a problem [2].

There are two types of SLR systems – isolated SLR and continuous SLR. In isolated SLR, the system is trained to recognize a single gesture. Each image is labelled to represent an alphabet, a digit, or some special gesture. Continuous SLR is different from isolated gesture classification. In continuous SLR, the system is able to recognize and translate whole sentences instead of a single gesture [16][17]. Even with all the research that has been done in SLR, many inadequacies need to be dealt with by further research. Some of the issues and challenges that need to be worked on are as follows [16][2][4][6].

• Isolated SLR methods need to do strenuous labeling for each word.

• Continuous SLR methods make use of isolated SLR systems as building blocks with temporal segmentation as pre-processing, which is non-trivial and unescapably proliferates errors into subsequent steps, and sentence synthesis as post-processing.

• Devices needed for data acquisition are costly, a cheap method is needed for SLR systems to be commercialized.

• Web camera is an alternative to higher specification camera but the image is blurred so, the quality is compromised.

• Data acquisition by sensors also has some issues e.g., noise, bad human manipulation, bad ground connection, etc.

• Vision-based methodologies introduce inaccuracies due to overlapping of hand and finger.

• Large datasets are not available.

• There are misconceptions about sign languages like sign language is same around the world, while sign language is based upon the spoken language.

• Indian Sign Language is communicated using hand gestures made by a single hand and double hands due to which there are two types of gestures representing the same thing.

In this paper, the dataset that will be used is created using Python and OpenCV with the help of a webcam. The SLR system that is being developed is a real-time detection system.

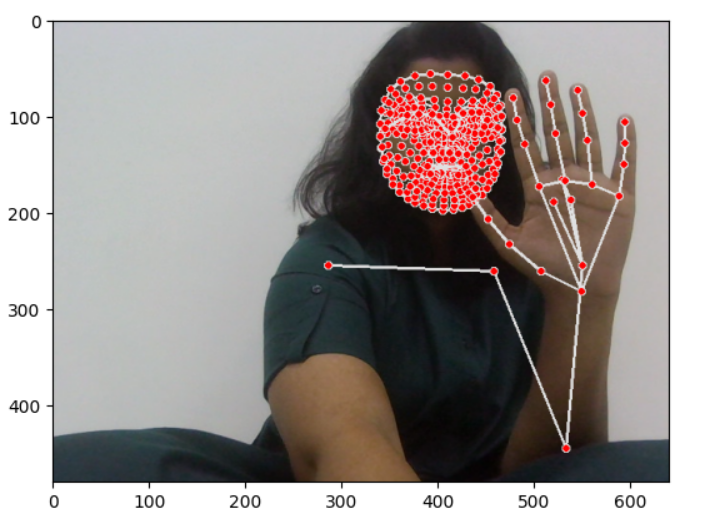
1. **Data Acquisition**

A real-time sign language detection system is being developed for Indian Sign Language. For data acquisition, images are captured by webcam using Python and OpenCV. OpenCV provides functions which are primarily aimed at the real-time computer vision. It accelerates the use of machine perception in commercial products and provides a common infrastructure for the computer vision-based applications. The OpenCV library has more than 2500 efficient computer vision and machine learning algorithms which can be used for face detection and recognition, object identification, classification of human actions, tracking camera and object movements, extracting 3D object models, and many more [18]. The created dataset is made up of signs representing alphabets in Indian Sign Language [19] as shown in Fig. 1. For every alphabet, 25 images are captured to make the dataset. The images are captured in every 2 seconds providing time to record gesture with a bit of difference every time and a break of five seconds are given between two individual signs, i.e., to change the sign of one alphabet to the sign of a different alphabet, five seconds interval is provided. The captured images are stored in their respective folder.

****

**Fig. 1.** Indian Sign Language Alphabets

For data acquisition, dependencies like cv2, i.e., OpenCV, os, time, and uuid have been imported. The dependency os is used to help work with file paths. It comes under standard utility modules of Python and provides functions for interacting with the operating systems. With the help of the time module in Python, time can be represented in multiple ways in code like objects, numbers, and strings. Apart from representing time, it can be used to measure code efficiency or wait during code execution. Here, it is used to add breaks between the image capturing in order to provide time for hand movements. The uuid library is used in naming the image files. It helps in the generation of random objects of 128 bits as ids providing uniqueness as the ids are generated on the basis of time and computer hardware.



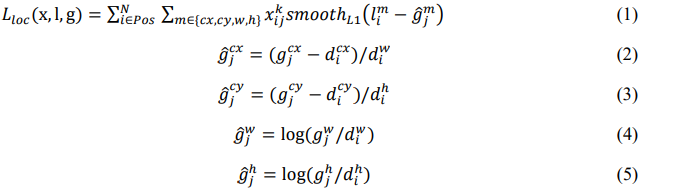
**Fig. 2.** Marking of LandMark Points

Once all the images have been captured, they are then one by one labelled using the LabelImg package. LabelImg is a free open-source tool for graphically labelling images. The hand gesture portion of the image is labelled by what the gesture in the box or the sign represents as shown in Fig. 2 and Fig. 3. On saving the labelled image, its XML file is created. The XML files have all the details of the images including the detail of the labelled portion. After labelling all the images, their XML files are available. This is used for creating the TF (TensorFlow) records. All the images along with their XML files are then divided into training data and validation data in the ratio of 80:20. From 25 images of an alphabet, 20 (80%) of them were taken and stored as a training dataset and the remaining 5 (20%) were taken and stored as validation dataset. This task was performed for all the images of all 26 alphabets.

1. **Methodology**

The proposed system is designed to develop a real-time sign language detector using a TensorFlow, MediaPipe, OpenCV and Keras and train it through transfer learning for the created dataset [20]. For data acquisition, images are captured by a webcam using Python and OpenCV following the procedure described under Section 3. Following the data acquisition, a labeled map is created which is a representation of all the objects within the model, i.e., it contains the label of each sign along with their id. The label map contains 26 labels, each one representing an alphabet. Each label has been assigned a unique id ranging from 1 to 26. This will be used as a reference to look up the class name. TF records of the training data and the testing data are then created using generate\_tfrecord which is used to train the TensorFlow . TF record is the binary storage format of TensorFlow. Binary files usage for storage of the data significantly impacts the performance of the import pipeline consequently, the training time of the model. It takes less space on a disk, copies fast, and can efficiently be read from the disk. The open-source framework, TensorFlow object detection API makes it easy to develop, train and deploy an object detection model. They have their framework called the TensorFlow detection model zoo which offers various models for detection that have been pre-trained on the COCO 2017 dataset.

After setting up and updating the configuration, the model was trained in 20000 steps. The hyper-parameter used during the training was to set up the number of steps in which the model will be trained which was set up to 20000 steps. During the training, the model has some losses as classification loss, regularization loss, and localization loss. The localization loss is mismatch. between the predicted bounding box correction and the true values. The formula of the localization loss [21] is given in Eq. (1) – (5).

****

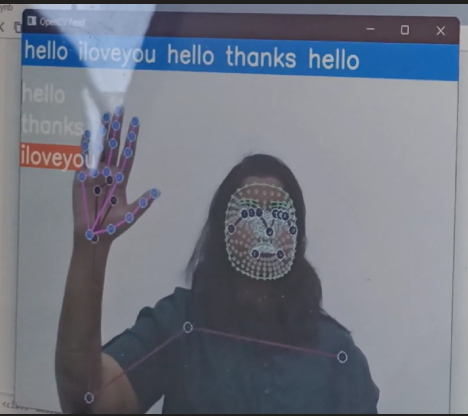
where, N is the number of the matched default boxes, l is the predicted bounding box, g is the ground truth bounding box, ĝ is the encoded ground truth bounding box and is the matching indicator between default box i and ground truth box j of category k.

The classification loss is defined as the softmax loss over multiple classes. The formula of the classification loss [21] is as Eq. (6)

****

where, is the softmax activated class score for default box I with category p, is matching g indicator between default box i and the ground truth box j of category p.

The different losses incurred during the experimentation are mentioned in the subsequent section. After training, the model is loaded from the latest checkpoint which makes it ready for real-time detection. After setting up and updating the configuration, the model will be ready for training. The trained model is loaded from the latest checkpoint which is created during the training of the model. This completes the model making it ready for real-time sign language detection. The real-time detection is done using OpenCV and webcam again. For, real-time detection, cv2, and NumPy dependencies are used. The system detects signs in realtime and translates what each gesture means into English as shown in Fig. 3. The system is tested in real-time by creating and showing it different signs. The confidence rate of each sign , i.e., how confident the system is in recognizing a sign is checked, noted, and tabulated for the result.



**Fig. 3.** Real-Time Sign Translation

1. **Experimental Evaluation**

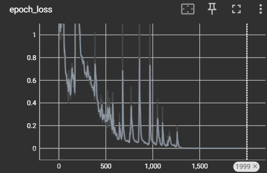
**5.1 Dataset and Experimental Setup**

The dataset is created for Indian Sign Language where signs are alphabets of the English language. The dataset is created following the data acquisition method described in Section 3.

The experimentation was carried out on a system with an Intel i5 7th generation 2.70 GHz processor, 8 GB memory and webcam (HP TrueVision HD camera with 0.31 MP and 640x480 resolution), running Windows 10 operating system. The programming environment includes Python (version 3.7.3), Jupyter Notebook, OpenCV (version 4.2.0), TensorFlow and Mediapipe.

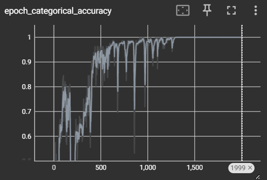
**5.2 Results and Discussion**

The developed system is able to detect Indian Sign Language alphabets in real-time. The system has been created using TensorFlow. It has been trained using transfer learning on the created dataset which contains 650 images in total, 25 images for each alphabet. The total loss incurred during the last part of the training, at 10,000 steps was 0.19, localization loss was 0.12, classification loss was 0.09, and regularization loss was 0.07 as shown in Fig. 4. Fig. 4 also shows that the lowest lost 0.17 was suffered at steps 9900.

****

**Fig. 4.** Loss Graph

The result of the system is based on the confidence rate and the average confidence rate of the system is 95.45%.The confidence rate of the system can be increased by increasing the size of the dataset which will boost up the recognition ability of the system. Thus, improving the result of the system and enhancing it.



**Fig. 5.**  Confidence Graph

The state-of-the-art method of the Indian Sign Language Recognition system achieved 93-96% accuracy [4]. Though being highly accurate, it is not a real-time SLR system. This issue is dealt with in this paper.

1. **Conclusion and Future Works**

Sign languages represent a visual mode of communication, utilizing hand movements, body gestures, and facial expressions. They play a crucial role in facilitating communication for individuals with disabilities, enabling them to convey emotions and thoughts effectively. However, a significant challenge arises from the limited widespread knowledge of sign languages among the general population, which hinders seamless communication.

To address this challenge, automated Sign Language Recognition (SLR) systems offer a promising solution. These systems can translate sign language gestures into commonly spoken languages, enhancing communication accessibility. This paper focuses on implementing SLR using the TensorFlow object detection API. Specifically, the system has been trained on a dataset comprising the Indian Sign Language alphabet, enabling it to accurately detect and interpret sign language gestures in real-time.

For data acquisition, the system employs cost-effective methods such as capturing images via a webcam using Python and OpenCV. This approach not only reduces costs but also simplifies the setup, making SLR technology more accessible for practical applications.

The developed system is showing an average confidence rate of 95.45%. Though the system has achieved a high average confidence rate, the dataset it has been trained on is small in size and limited. In the future, the dataset can be enlarged so that the system can recognize more gestures. The TensorFlow model that has been used can be interchanged with another model as well. The system can be implemented for different sign languages by changing the dataset.

**References**

1. Kapur, R.: The Types of Communication. MIJ. 6, (2020)
2. Suharjito, Anderson, R., Wiryana, F., Ariesta, M.C., Kusuma, G.P.: Sign Language Recognition Application Systems for Deaf-Mute People: A Review Based on Input Process-Output. Procedia Comput. Sci. 116, 441–448 (2017). https://doi.org/10.1016/J.PROCS.2017.10.028.
3. Konstantinidis, D., Dimitropoulos, K., Daras, P.: Sign language recognition based on hand and body skeletal data. 3DTV-Conference. 2018-June, (2018). <https://doi.org/10.1109/3DTV.2018.8478467>.
4. Dutta, K.K., Bellary, S.A.S.: Machine Learning Techniques for Indian Sign Language Recognition. Int. Conf. Curr. Trends Comput. Electr. Electron. Commun. CTCEEC 2017. 333–336 (2018). <https://doi.org/10.1109/CTCEEC.2017.8454988>.
5. Bragg, D., Koller, O., Bellard, M., Berke, L., Boudreault, P., Braffort, A., Caselli, N., Huenerfauth, M., Kacorri, H., Verhoef, T., Vogler, C., Morris, M.R.: Sign Language Recognition, Generation, and Translation: An Interdisciplinary Perspective. 21st Int. ACM SIGACCESS Conf. Comput. Access. (2019). <https://doi.org/10.1145/3308561>.
6. Rosero-Montalvo, P.D., Godoy-Trujillo, P., Flores-Bosmediano, E., Carrascal-Garcia, J., 12 Otero-Potosi, S., Benitez-Pereira, H., Peluffo-Ordonez, D.H.: Sign Language Recognition Based on Intelligent Glove Using Machine Learning Techniques. 2018 IEEE 3rd Ecuador Tech. Chapters Meet. ETCM 2018. (2018). <https://doi.org/10.1109/ETCM.2018.8580268>.
7. Zheng, L., Liang, B., Jiang, A.: Recent Advances of Deep Learning for Sign Language Recognition. DICTA 2017 - 2017 Int. Conf. Digit. Image Comput. Tech. Appl. 2017- Decem, 1–7 (2017). <https://doi.org/10.1109/DICTA.2017.8227483>.
8. Rautaray, S.S.: A Real Time Hand Tracking System for Interactive Applications. Int. J. Comput. Appl. 18, 975–8887 (2011).
9. Zhang, Z., Huang, F.: Hand tracking algorithm based on super-pixels feature. Proc. - 2013 Int. Conf. Inf. Sci. Cloud Comput. Companion, ISCC-C 2013. 629–634 (2014). <https://doi.org/10.1109/ISCC-C.2013.77>.
10. Lim, K.M., Tan, A.W.C., Tan, S.C.: A feature covariance matrix with serial particle filter for isolated sign language recognition. Expert Syst. Appl. 54, 208–218 (2016). <https://doi.org/10.1016/J.ESWA.2016.01.047>.
11. Lim, K.M., Tan, A.W.C., Tan, S.C.: Block-based histogram of optical flow for isolated sign language recognition. J. Vis. Commun. Image Represent. 40, 538–545 (2016). <https://doi.org/10.1016/J.JVCIR.2016.07.020>.
12. Gaus, Y.F.A., Wong, F.: Hidden Markov Model - Based gesture recognition with overlapping hand-head/hand-hand estimated using Kalman Filter. Proc. - 3rd Int. Conf. Intell. Syst. Model. Simulation, ISMS 2012. 262–267 (2012). <https://doi.org/10.1109/ISMS.2012.67>.
13. Nikam, A.S., Ambekar, A.G.: Sign language recognition using image based hand gesture recognition techniques. Proc. 2016 Online Int. Conf. Green Eng. Technol. IC-GET 2016. (2017). <https://doi.org/10.1109/GET.2016.7916786>.
14. Mohandes, M., Aliyu, S., Deriche, M.: Arabic sign language recognition using the leap motion controller. IEEE Int. Symp. Ind. Electron. 960–965 (2014). <https://doi.org/10.1109/ISIE.2014.6864742>.
15. Enikeev, D.G., Mustafina, S.A.: Sign language recognition through Leap Motion controller and input prediction algorithm. J. Phys. Conf. Ser. 1715, 012008 (2021). <https://doi.org/10.1088/1742-6596/1715/1/012008>.
16. Huang, J., Zhou, W., Zhang, Q., Li, H., Li, W.: Video-based Sign Language Recognition without Temporal Segmentation.
17. Cui, R., Liu, H., Zhang, C.: Recurrent convolutional neural networks for continuous sign language recognition by staged optimization. Proc. - 30th IEEE Conf. Comput. Vis. Pattern Recognition, CVPR 2017. 2017-Janua, 1610–1618 (2017). <https://doi.org/10.1109/CVPR.2017.175>.
18. About - OpenCV.
19. Poster of the Manual Alphabet in ISL | Indian Sign Language Research and Training Center 14 ( ISLRTC), Government of India.
20. Transfer learning and fine-tuning | TensorFlow Core.
21. Wu, S., Yang, J., Wang, X., Li, X.: IoU-balanced Loss Functions for Single-stage Object Detection. (2020).