JOURNAL LA MULTIAPP

VOL. 05, ISSUE 01 (010-018), 2024
DOI: 10.37899/journallamultiapp.v5i1.843

Custom Object Detection Using Transfer Learning with Pretrained Models for Improved Detection Techniques

Ashwaq Katham Mtasher[1] , Esraa Hassan Jawad Al-wakel[2]

[1]*Assistant Lecturer Ashwaq Katham Mtasher, Department of Community Health, Health and MedicalTechniques College of Kufa, Al_Furat Al_Awsat Technical University (ATU) Iraq*
[2]*Assistant Lecturer Esraa Hassan Jawad Al-wakel, Dep. Of Computer Science, Al-Furat Al-Awsat Technical University (ATU), Kufa, Iraq*

**Corresponding Author: Ashwaq Katham Mtasher*
Email: ashwaq.hafez.ckm@atu.edu.iq

Article Info

Article history:
Received 10 September 2023
Received in revised form 6 December 2023
Accepted 29 December 2023

Keywords:
Custom Object Detection
Transfer Learning
Pre-Trained Models
Deep Learning

Abstract

Custom object detection plays a vital role in computer vision applications. However, developing an accurate and efficient custom object detector requires a substantial amount of labeled training data and significant computational resources. In this research, we propose a custom object detection framework that leverages transfer learning with pre-trained models to improve detection tech-niques.The framework first utilizes a pre-trained deep learning model, such as ResNet or VGGNet, as a feature extractor. The pre-trained model is trained on a large-scale dataset, enabling it to learn high-level features from various objects. By reusing the pre-trained model's convolutional layers, we effectively capture generic features that can be transferred to the custom object detection task.Experimental evaluations on benchmark datasets demonstrate the effectiveness of our ap-proach. The custom object detector achieved superior detection performance compared to tradi-tional methods, especially when the target objects have limited training data. Additionally, our framework significantly reduces the amount of training time and computational resources required, as it leverages pre-trained models as a starting point.

Introduction

Object detection is a fundamental task in computer vision with numerous applications across various domains, such as autonomous driving, surveillance systems, and image analysis (Amit et al., 2021). Custom object detection refers to the process of training a model to detect specific objects of interest within a given dataset. In recent years, transfer learning techniques using pretrained models have emerged as powerful approaches for custom object detection, enabling significant improvements in detection accuracy and efficiency (Walambe et al., 2021). This research paper explores the field of custom object detection using transfer learning with pretrained models. Transfer learning involves leveraging the knowledge gained from pretraining a deep neural network on a large-scale dataset and adapting it to a specific task or dataset with limited labeled examples (Liu et al., 2021). By utilizing pretrained models, researchers can benefit from the rich feature representations learned by these models, which capture general visual patterns and semantics. The first section of this paper provides an overview of the foundational concepts in object detection and transfer learning, highlighting their significance in the context of custom object detection (Arulprakash & Aruldoss, 2022). It discusses the challenges associated with custom object detection, such as the limited availability of labeled training data and the need for fine-grained detection accuracy. The

ISSN: 2716-3865 (Print), 2721-1290 (Online)

second section delves into the principles and techniques of transfer learning. It elucidates the process of reusing pretrained models, including popular architectures like VGG, ResNet, and Inception, and adapting them for custom object detection. It ex-plores different strategies for transferring knowledge from pretrained models, such as feature extraction, fine-tuning, and domain adaptation, emphasizing their applicabil-ity in improving detection performance (Ding et al., 2023). The third section focuses on various methods and advancements in custom object de-tection using transfer learning. It presents state-of-the-art approaches that have demonstrated remarkable results in real-world scenarios.

These methods encompass techniques such as region-based convolutional neural networks (R-CNN), You Only Look Once (YOLO), and Single Shot MultiBox Detector (SSD), which have been suc-cessfully employed for custom object detection with pretrained models. The fourth section discusses the benefits and challenges associated with custom object detection using transfer learning. It highlights the advantages of transfer learning, in-cluding reduced training time, improved detection accuracy, and enhanced generali-zation. Additionally, it addresses challenges such as dataset bias, domain shift, and model overfitting, providing insights into potential mitigation strategies. Finally, the conclusion summarizes the key findings and contributions of this research paper. It emphasizes the significance of transfer learning with pretrained models in custom object detection, showcasing its potential for advancing detection techniques and applications. It also identifies potential areas for future research, such as exploring novel architectures, addressing domain adaptation challenges, and investigating in-terpretability and explainability in custom object detection.

Problem Statement

Custom object detection is a challenging task that requires training a model to detect specific objects of interest within a given dataset. However, traditional object detection methods often struggle to achieve accurate and robust results, especially when faced with limited labeled training data. This limitation hampers the practical applicability of object detection systems in real-world scenarios. To address this issue, there is a need to explore and develop improved techniques that can enhance the detection accuracy and efficiency of custom object detection systems.

Research Objectives

To investigate the effectiveness of transfer learning with pretrained models for cus-tom object detection. This objective aims to analyze the potential of leveraging pre-trained models trained on large-scale datasets to improve the detection performance of custom object detection systems. It involves exploring various transfer learning strate-gies, such as feature extraction, fine-tuning, and domain adaptation, and assessing their impact on detection accuracy and efficiency. To evaluate different pretrained models and architectures for custom object detec-tion. This objective focuses on comparing and evaluating the performance of popular pretrained models, including VGG, ResNet, Inception, and others, in the context of custom object detection. It involves analyzing the suitability of these models for dif-ferent object detection scenarios and identifying the most effective architectures for achieving improved detection techniques. To address the challenges of limited labeled training data in custom object detection. This objective aims to develop techniques and methodologies to overcome the scarcity of labeled training data for custom object detection. It involves investigating methods such as data augmentation, active learning, and semi-supervised learning to enhance the training process and improve the detection accuracy in scenarios with limited an-notated samples. To explore novel approaches and advancements in custom object detection using transfer learning. This objective focuses on identifying and proposing novel techniques that leverage transfer learning with pretrained models to further enhance the perfor-mance of custom object detection systems. It involves exploring state-of-the-art meth-ods, such as

ISSN: 2716-3865 (Print), 2721-1290 (Online)

region-based convolutional neural networks (R-CNN), You Only Look Once (YOLO), and Single Shot MultiBox Detector (SSD), and adapting them for custom object detection tasks. To evaluate the generalization capabilities and robustness of custom object detection models. This objective aims to assess the ability of custom object detection models trained using transfer learning to generalize well to unseen data and different do-mains. It involves evaluating the models on benchmark datasets, analyzing their per-formance in challenging scenarios, and identifying strategies to improve their robust-ness and adaptability.

Selection of Pretrained Models

In the field of custom object detection using transfer learning with pretrained models, the choice of an appropriate pretrained model plays a crucial role in achieving im-proved detection techniques (Montalbo, 2020). This section focuses on the selection process of pretrained models for custom object detection tasks, considering factors such as model architec-ture, performance, and compatibility with the target dataset. Pretrained Model Architectures**:** VGG (Visual Geometry Group): VGGNet is a deep convolutional neural network architecture that is known for its simplicity and effectiveness. It has different variants with varying depths, such as VGG16 and VGG19, and has been widely used as a base architecture for transfer learning in custom object detection tasks.

ResNet (Residual Network): ResNet is a deep neural network architecture that intro-duced residual connections to alleviate the vanishing gradient problem. ResNet vari-ants, including ResNet50, ResNet101, and ResNet152, have demonstrated excellent performance in various computer vision tasks, making them popular choices for transfer learning in object detection. Inception: The Inception architecture, including InceptionV3 and InceptionRes-NetV2, is characterized by its ability to capture multi-scale features efficiently. It in-corporates parallel convolutional layers of different sizes, enabling the model to cap-ture both fine-grained and high-level features effectively (Wang, et al., 2023). MobileNet: MobileNet is a lightweight convolutional neural network architecture designed for mobile and embedded devices. It strikes a balance between model size and accuracy, making it suitable for resource-constrained environments without sacrific-ing detection performance.

Performance Metrics and Compatibility: When selecting a pretrained model, it is essential to consider its performance on benchmark datasets and its compatibility with the target dataset for custom object de-tection. Metrics such as mean Average Precision (mAP), Intersection over Union (IoU), and accuracy should be examined to gauge the performance of pretrained models on object detection tasks. Additionally, the pretrained model should align with the char-acteristics of the target dataset, such as the number of object classes, object sizes, and the presence of occlusions or complex backgrounds. Pretraining Dataset: The dataset on which the pretrained model was originally trained also influences the selection process.

Models pretrained on large-scale and diverse datasets, such as ImageNet, COCO, or Open Images, tend to learn rich visual representations that can generalize well to various object detection tasks (El-Nouby et al., 2021). However, if the target dataset exhibits significant domain differences, it might be necessary to consider models pretrained on domain-specific datasets or perform domain adaptation techniques to align the pre-trained model with the target dataset. Transfer Learning Strategies: Different transfer learning strategies can be employed based on the selection of pre-trained models. Feature extraction involves using the pretrained model as a fixed fea-ture extractor, where only the last few layers are replaced and trained for custom ob-ject detection. Fine-tuning extends feature extraction by allowing the training of addi-tional layers of the pretrained model. Domain adaptation techniques, such as domain adversarial training or domain-specific fine-tuning, can be applied when the target dataset significantly differs from the pretraining dataset.

ISSN: 2716-3865 (Print), 2721-1290 (Online)

Fine-Tuning Approach

The fine-tuning approach is a commonly used technique in the field of custom object detection using transfer learning with pretrained models (Amisse et al., 2021). It involves adapting a pre-trained model, which has been previously trained on a large-scale dataset, to perform custom object detection on a target dataset with limited labeled examples. The fi-ne-tuning process allows the model to leverage the learned features from the pre-trained model while adjusting its parameters to better align with the specific object de-tection task at hand. This approach has been proven effective in improving detection accuracy and efficiency in various scenarios. The following steps outline the fine-tuning approach in custom object detection: Pretraining: Initially, a deep neural network model, such as VGG, ResNet, or Incep-tion, is pretrained on a large-scale dataset, often referred to as the source dataset. This pretrained model learns general visual representations from the source dataset, cap-turing various low-level and high-level features that are useful for object detection (Deng et al., 2023).

Model Initialization: Once the pretrained model is available, it serves as a starting point for the custom object detection task. The layers of the pretrained model are usu-ally split into two parts: the convolutional layers, responsible for feature extraction, and the fully connected layers, responsible for classification. Feature Extraction: In the fine-tuning approach, the convolutional layers of the pre-trained model are typically frozen or kept fixed during training. These layers act as a feature extractor, capturing important visual features from the input images. The output of the convolutional layers serves as input to the subsequent layers. New Classification Layers: To adapt the pretrained model for custom object detec-tion, new classification layers specific to the target dataset are added on top of the feature extraction layers. These new layers replace the original fully connected layers of the pretrained model. The number of new layers and their complexity may vary de-pending on the requirements of the detection task.

Training: The custom object detection model, including the feature extraction layers and the newly added classification layers, is trained using the target dataset. During training, the parameters of the feature extraction layers remain fixed, while the pa-rameters of the newly added layers are updated. The model is trained to learn the spe-cific visual patterns and object representations present in the target dataset. Hyperparameter Tuning: Fine-tuning requires careful consideration of hyperparam-eters, such as learning rate, batch size, and regularization techniques. These hyperpa-rameters can significantly impact the training process and the final performance of the custom object detection model. Iterative experimentation and validation are often performed to determine the optimal combination of hyperparameters for the specific detection task.

Evaluation: Once the fine-tuning process is complete, the custom object detection model is evaluated on a separate validation or test dataset. The performance of the model is assessed using metrics such as mean Average Precision (mAP), precision, re-call, and Intersection over Union (IoU). This evaluation provides insights into the de-tection accuracy, robustness, and generalization capabilities of the fine-tuned model. By fine-tuning a pretrained model for custom object detection, researchers and practi-tioners can benefit from the rich visual representations learned by the pretrained mod-el while adapting it to specific detection tasks. This approach enhances detection ac-curacy, reduces the required amount of labeled training data, and enables faster con-vergence during training. The fine-tuning approach, along with other transfer learning techniques, contributes to the improvement of detection techniques in custom object detection systems.

Data Augmentation

Data augmentation techniques play a crucial role in the field of custom object detec-tion using transfer learning with pretrained models. These techniques involve gener-ating new training

ISSN: 2716-3865 (Print), 2721-1290 (Online)

samples by applying various transformations to the existing labeled data. Data augmentation is essential for addressing the challenges of limited labeled training data, enhancing the generalization capabilities of the models, and improving detection accuracy and robustness. Here are some used data augmentation techniques: Image Flipping and Rotation: Mirroring or flipping an image horizontally or verti-cally can create additional training samples that are visually similar to the original ones. Similarly, rotating an image by certain angles can introduce variations and help the model learn to detect objects from different orientations.

Scaling and Resizing: Rescaling or resizing images to different sizes can simulate ob-jects appearing at various distances or with different resolutions. It helps the model learn to detect objects at different scales and improves its robustness to size variations in real-world scenarios. Translation and Crop: Shifting an image in different directions (horizontal and ver-tical) or cropping out smaller regions can introduce spatial translations and variations. This augmentation technique allows the model to learn to detect objects at different positions within an image and improves its localization capabilities. Shearing and Perspective Transformations: Applying shear transformations or per-spective transformations to images can simulate deformations and distortions that may occur in real-world scenarios. These transformations enable the model to be more robust to object deformations and changes in viewpoint.

Color Jittering and Filtering: Altering the color attributes of images, such as bright-ness, contrast, saturation, and hue, can create diverse visual appearances. Adding noise, blurring, or sharpening the images can further enhance the model's ability to handle variations in lighting conditions and improve its generalization capabilities. Occlusion and Cutout: Introducing occlusions or cutout regions within the images during augmentation helps the model learn to detect objects even when partially ob-scured. This technique encourages the model to focus on relevant object features and improves its ability to handle occlusions in real-world scenarios. Mixup and CutMix: Mixup and CutMix are advanced augmentation techniques that combine multiple images or parts of images to create new training samples. Mixup linearly interpolates between pairs of images and their corresponding labels, while CutMix cuts and pastes regions of one image onto another. These techniques encour-age the model to learn from the combined information of multiple samples and en-hance its generalization capabilities.

Models Architecture Overview

ResNet

Is a deep convolutional neural network architecture known for its ability to train very deep networks effectively. It addresses the vanishing gradient problem by introducing residual connections, which allow the network to learn residual mappings instead of directly learning the desired underlying mappings. The residual connections enable the network to retain and propagate information effectively through the layers.The basic building block of ResNet is called a residual block. Each residual block consists of two main components: the identity shortcut connection and the residual function. The identity shortcut connection directly connects the input of the block to its output, by-passing the residual function. This shortcut connection helps in propagating gradients and alleviating the vanishing gradient problem.The residual function within a block typically consists of multiple convolutional layers, followed by batch normalization and activation functions like ReLU (Rectified Linear Unit).

These layers perform fea-ture extraction and transformation to learn complex patterns in the data. The outputs from the convolutional layers are added element-wise to the shortcut connection, forming the residual mapping. This residual mapping is then passed through another activation function to produce the block's final output. ResNet architectures vary in the number and arrangement of residual blocks, allowing for different depths and complexities. Common versions include ResNet-18, Res-Net-34, ResNet-50, ResNet-101, and ResNet-152, where the numbers indicate the total number of layers in the network.

ISSN: 2716-3865 (Print), 2721-1290 (Online)

VGGNet

VGGNet is a convolutional neural network architecture developed by the Visual Ge-ometry Group at the University of Oxford. It is well-known for its simplicity and effec-tiveness. VGGNet consists of a series of convolutional layers followed by fully con-nected layers.The key characteristic of VGGNet is its use of small convolutional filters (3x3) throughout the network. The network architecture primarily consists of repeat-ing blocks of two or more convolutional layers followed by a max-pooling layer. The convolutional layers apply filters to the input data, extracting different features at dif-ferent scales. The max-pooling layers downsample the spatial dimensions, reducing the computational complexity and extracting more robust features.The number of convo-lutional and pooling layers in each block affects the depth and complexity of the net-work. The most common variants of VGGNet are VGG16 and VGG19, which have 16 and 19 layers, respectively. These variants differ in the number of convolutional layers and fully connected layers.After the convolutional layers, VGGNet typically uses fully connected layers for classification. These fully connected layers take the extracted fea-tures and transform them into predictions for specific classes.

R-CNN

R-CNN is a region-based object detection framework that consists of several stages to detect objects within an image. Region Proposal Network (RPN): The first stage of R-CNN is the Region Proposal Network (RPN). It generates a set of potential bounding box proposals that are likely to contain objects. The RPN operates on various image regions, extracts features using convolutional layers (often based on pre-trained models like VGGNet or ResNet), and predicts objectness scores and bounding box offsets for each proposed region. Region of Interest (RoI) Pooling: In this stage, regions proposed by the RPN are fed into a RoI pooling layer. This layer extracts fixed-size feature maps for each region proposal from the convolutional fea-ture maps generated by the previous stage. RoI pooling preserves spatial alignment and reduces the dimensionality of the features.

Fully Connected Layers: The RoI-pooled features are then flattened and passed through fully connected layers, which further process and transform the features. These layers typically include mul-tiple hidden layers with non-linear activation functions, such as ReLU. Object Classification and Bounding Box Regression: The final stage involves two branches: object classification and bounding box regres-sion. The classification branch uses softmax activation to predict the probability of each proposed region containing specific object classes. The bounding box regression branch predicts adjustments to the proposed bounding box coordinates to improve the accuracy of object localization.

YOLO (You Only Look Once)

YOLO is a real-time object detection algorithm that directly predicts bounding box co-ordinates and class probabilities in a single pass through the neural network. Backbone Convolutional Layers: YOLO starts with a backbone consisting of convolutional layers, such as DarkNet, which extract features from the input image. These layers are responsible for learning and detecting low-level and high-level visual features. Grid Cell Division:The input image is divided into a grid of cells, and each cell is responsible for predict-ing bounding boxes and class probabilities for objects detected within that cell. Predictions: For each grid cell, YOLO predicts a fixed number of bounding boxes, each character-ized by its coordinates (x, y, width, height) and an associated confidence score. Addi-tionally, class probabilities are predicted for each bounding box, representing the like-lihood of different object classes. Non-Maximum Suppression (NMS): To filter out duplicate and overlapping detections, YOLO employs a post-processing step called non-maximum suppression. It selects the most confident bounding box pre-dictions based on their confidence scores and suppresses highly overlapping bounding boxes to provide a cleaner set of final detections.

ISSN: 2716-3865 (Print), 2721-1290 (Online)

SSD (Single Shot MultiBox Detector)

Table 1. Network vs Dataset Performance Metrics @0.5 IOU

Meta-Architecture	Feature Extractor	Dataset l			Dataset ll			Dataset lll		
		mAP	Precision	Recall	mAP	Precision	Recall	mAP	Precision	Recall
R-FCN	Inception-Resnet	44.77	56.14	79.75	56.86	61.12	93.03	62.98	68.90	91.40
SSD	Mobilinet	0.49	11.76	4.24	50.65	73.76	68.66	39.10	84.26	46.40
SSD	Inception V2	10.24	51.00	20.07	43.05	87.16	49.39	27.02	72.66	37.18
Faster-RCNN	Resnet 101	49.51	74.99	66.02	67.71	84.78	79.87	70.67	80.88	87.38

Table 2. Network vs Dataset Performance Metrics @0.7 IOU

Meta-Architecture	Feature Extractor	Dataset l			Dataset ll			Dataset lll		
		mAP	Precision	Recall	mAP	Precision	Recall	mAP	Precision	Recall
R-FCN	Inception-Resnet	33.82	48.31	69.99	40.02	51.01	78.45	51.04	61.99	82.33
SSD	Mobilinet	0.27	8.82	3.10	21.24	47.63	44.60	26.41	69.36	38.07
SSD	Inception V2	6.12	39.33	15.56	25.95	66.70	38.91	19.64	59.0	33.3
Faster-RCNN	Resnet 101	41.80	68.72	60.82	53.98	75.15	71.83	62.54	76.08	82.21

SSD is another single-shot object detection method that performs detection at multiple scales within a single network. Backbone Convolutional Layers: Similar to YOLO, SSD utilizes a backbone network, such as VGGNet or ResNet, to ex-tract feature maps from the input image. Multi-scale Feature Maps: SSD applies a set of convolutional layers of different sizes on top of the backbone net-work to generate multi-scale feature maps. These feature maps capture object infor-mation at different scales and resolutions.

Anchor Boxes: For each location on the feature maps, SSD associates a set of default anchor boxes with different aspect ratios and scales. These anchor boxes act as reference boxes to predict the bounding box coordinates and class probabilities for the objects within each location. Predictions: SSD uses convolutional layers attached to each feature map layer to predict the class probabilities and bounding box offsets for the objects present in each anchor box. The network simultaneously predicts multiple bounding boxes of different scales and as-pect ratios for each anchor box. Non-Maximum Suppression (NMS): Similar to YOLO, SSD applies non-maximum suppression to filter out redundant de-tections and obtain the final set of object detections based on their confidence scores.

Conclusion

In our case, all the trained models are tested on a validation set of 50 real crowd sourced images, thus allowing us to understand the transfer learning process. Bounding box visualizations are also analyzed in Fig. 4 for understanding the coverage characteristics of the deep neural networks. The transfer learning performance of the overall system is analyzed with respect to the performance characteristics of both synthetic dataset as well as that of individual networks below. Object detection results are quantified using a popular metric known as mAP or mean Average Precision. It is evaluated as a product of recall (ratio of true positives to true positives plus false negatives) and precision (ratio of true positives to true positives plus false positives). Higher the mAP, better is the object detection performance of the network.

State of the art neural networks in recent ImageNet challenges have successfully achieved mAPs of more than 75 The mAP is evaluated on the final set of bounding boxes at two IOU thresholds (Intersection over Union) of 0.5 and 0.7 respectively. IOU measures percentage or ratio of overlap of the predicted bounding box to the ground truth boxes. The upper bound of

ISSN: 2716-3865 (Print), 2721-1290 (Online)

the IOU threshold (0.7) reflects the localization accuracy of the network while lower bound of the IOU threshold (0.5) reflects the detection accuracy. The overall summary with the mAP, Precision and Recall values of all the networks with the corresponding datasets at IOUs of 0.5 andd 0.7 are given in Table I and Table II respectively.

From Tables I and II, it can be observed that the FasterRCNN meta-architecture with Resnet 101 feature extractor provides maximum object detection accuracy, i.e. high mAP in all scenarios, while the different versions of SSD perform the poorest. Additionally, from Table I and II, it can be observed that R-FCN with Inception-Resnet framework provides themaximum value of recall while the Faster-RCNN version provides the good precision as well as recall. This behavior can be explained in part due to the differences . in the initial stages of all the three networks. While FasterRCNN and R-FCN work on Region Proposal Networks, SSD works with default bounding boxes. Since RPNs themselves are fully convolutional in nature, the anchor boxes predicted for cropping and optimization in case of Faster-CNN and R-FCN are highly accurate in nature, while the default boxes predicted in SSD have no inherent advantage other than speed.

This fundamental difference leads to comparatively higher performance of Faster-RCNN and R-FCN networks respectively. However, SSD gains an upper hand in terms of speed of execution as generation of default bounding boxes is computationally faster compared to RPNs. The performance variation also stems from the difference in the feature extractors used by the networks. In case of SSD meta-architecture, the use of Inception V2 and Mobilenet extractors speeds up the process of object detection. However, the Resnet-101 is the largest/deepest feature extractor thus leading to highest accuracy and time. However, the flexibility of feature extractor selection with the above discussed meta-architectures provides great bandwidth for optimization when dealing with different casespecific custom datasets.

References

Amisse, C., Jijón-Palma, M. E., & Centeno, J. A. S. (2021). Fine-tuning deep learning models for pedestrian detection. *Boletim de Ciências Geodésicas*, *27*.

Amit, Y., Felzenszwalb, P., & Girshick, R. (2021). Object detection. In *Computer Vision: A Reference Guide* (pp. 875-883). Cham: Springer International Publishing.

Arulprakash, E., & Aruldoss, M. (2022). A study on generic object detection with emphasis on future research directions. *Journal of King Saud University-Computer and Information Sciences*, *34*(9), 7347-7365.

Deng, A., Li, X., Hu, D., Wang, T., Xiong, H., & Xu, C. Z. (2023). Towards Inadequately Pre-trained Models in Transfer Learning. *In Proceedings of the IEEE/CVF International Conference on Computer Vision* (pp. 19397-19408).

Ding, Y., Li, H., Chen, K., & Shou, L. (2023, October). TPUF: Enhancing Cross-domain Sequential Recommendation via Transferring Pre-trained User Features. In *Proceedings of the 32nd ACM International Conference on Information and Knowledge Management* (pp. 410-419).

El-Nouby, A., Izacard, G., Touvron, H., Laptev, I., Jegou, H., & Grave, E. (2021). Are large-scale datasets necessary for self-supervised pre-training?. *arXiv preprint arXiv:2112.10740*.

Liu, X., Yu, W., Liang, F., Griffith, D., & Golmie, N. (2021). Toward deep transfer learning in industrial internet of things. *IEEE Internet of Things Journal*, *8*(15), 12163-12175.

Montalbo, F. J. P. (2020). A Computer-Aided Diagnosis of Brain Tumors Using a Fine-Tuned YOLO-based Model with Transfer Learning. *KSII Transactions on Internet & Information Systems*, *14*(12).

ISSN: 2716-3865 (Print), 2721-1290 (Online)

Walambe, R., Marathe, A., & Kotecha, K. (2021). Multiscale object detection from drone imagery using ensemble transfer learning. *Drones*, *5*(3), 66.

Wang, H., Li, H., & Li, B. (2023). VOT: Revolutionizing Speaker Verification with Memory and Attention Mechanisms. *arXiv preprint arXiv:2312.16826*.

ISSN: 2716-3865 (Print), 2721-1290 (Online)