

VISION TRANSFORMER FRAMEWORKS FOR VISUAL RECOGNITION: A CRITICAL ANALYSIS

¹A. SRILEKHA, ²A. UDAY KIRAN, ³B. SUDEEPTHI, ⁴G. VINESH SHANKER, ⁵CH. BHANU PRAKESH,

¹UG Student, ²Assistant Professor, ³UG Student, ⁴Assistant Professor, ⁵UG Student,

Department of CSE

CMR Technical Campus Hyderabad, Telangana, India

Srilekhaakkapally@gmail.com, davu.sandhya@gmail.com, enugalapavithra@gmail.com, gogikarvinesh@gmail.com,
vyshnavibellary1@gmail.com

Abstract: The transformer is a type of neural network that was first used in the field of natural language processing. It is based on the self-attention mechanism. The transformer is really good at representing things so researchers are trying to use it for computer vision tasks.

In visual benchmarks models that use the transformer do as well as or better than other types of networks like convolutional and recurrent neural networks. The transformer does well and does not need special vision rules so it is getting a lot of attention from the computer vision community. We look at the transformer models for vision in this paper. We put them into categories like the backbone network, high and mid level vision, low level vision and video processing. We also look at ways to make the transformer more efficient so it can be used on devices. We take a look at the self-attention mechanism, in computer vision because it is a big part of the transformer. At the end of this paper we talk about the challenges. Give some ideas for where to go next with vision transformers. The transformer is a deal and people are still figuring out how to use it for computer vision. The transformer is going to keep getting better and better.

IndexTerms - Vision Transformer, Visual Recognition, Deep Learning, Image Classification, Object Detection.

I. INTRODUCTION

Deep neural networks are the structure in today's artificial intelligence systems. Artificial intelligence systems like these need types of networks for different tasks. For example the multi-layer perceptron is a type of network. Neural networks like this are made up of layers and nonlinear activations stacked together. Convolutional neural networks are used for processing images. These convolutional neural networks use layers. Pooling layers. Recurrent neural networks are used for processing data or time series data. The Transformer is a type of network. The Transformer uses the self-attention mechanism to extract features. The Transformer has potential for use in intelligence applications. The Transformer was first used for natural language processing tasks. The Transformer achieved improvements.

The Transformer was first used for machine translation and English constituency parsing tasks. Then a new language representation model called BERT was introduced. BERT pre-trains a Transformer on text. The BERT system considers the context of each word. The BERT system achieved performance on natural language processing tasks. A big model called GPT-3 was trained on a lot of text data. The GPT-3 model performed well on natural language tasks without needing any training.

The BERT system and the GPT-3 model are examples of Transformer-based models that have achieved successes in natural language processing. People were inspired by how the Transformer models worked for natural language processing so they tried using the Transformer models for computer vision tasks.

In computer vision convolutional neural networks are considered the component. Now the Transformer is showing that it is an alternative to convolutional neural networks. A sequence Transformer was trained to predict pixels. It achieved results to convolutional neural networks on image classification tasks. Another vision Transformer model is ViT. ViT applies a Transformer directly to sequences of image patches to classify the image. ViT has achieved state-of-the-art performance on image recognition benchmarks.

The Transformer has been used to address vision problems. These include object detection, semantic segmentation, image processing and video understanding. Due to its performance many researchers are proposing Transformer-based models for improving visual tasks.

Because of the increase in the number of Transformer-based vision models it is becoming difficult to keep up with the progress. A survey of the existing works would be beneficial for the community. In this paper we provide an overview of the advances in vision Transformers. We discuss the directions for improvement.

We categorize the Transformer models by their application scenarios. The main categories include backbone network, high-level vision, mid-level vision, low-level vision and video processing. High-level vision deals with the interpretation and use of what's seen in the image. Mid-level vision deals with how this information is organized into objects and surfaces. We treat high-level vision and mid-level vision as a category. A few examples of Transformer models that address these tasks include DETR for object detection and Max-Deep Lab for segmentation.

Low-level image processing deals with extracting descriptions from images. Typical applications of low-level image processing include super-resolution, image denoising and style transfer. Video processing is a part of computer vision. The Transformer is well suited for use on video tasks. It is beginning to perform on par with networks and recurrent neural networks.

We survey the works associated with Transformer-based models to track the progress in this field. The development timeline of the vision Transformer will undoubtedly have milestones in the future. The rest of the paper is organized as follows. We discuss the formulation of the standard Transformer and the self-attention mechanism. Then we summarize the vision Transformer models on backbone high-level vision, mid-level vision, low-level vision and video tasks.

We also describe Transformer methods. In the section we give our conclusion. Discuss several research directions and challenges. Due to the page limit we describe the methods of the Transformer in natural language processing in the material. We also review the self-attention mechanism for computer vision in the material. In this survey we mainly include the works since there are preprinted works, on arXiv.

II. LITERATURE SURVEY

A. The Perceptron, a machine that can see and recognize things Project Para

Point cloud based place recognition is very important for robotics. In this paper we talk about a way to do point cloud place recognition. We use a method called aggregation. This method uses information about the structure of things to help recognize places. First we combine two types of features: learned features and handcrafted features. Then we use these combined features to extract and aggregate features. We do this by looking at how dense things are where they are in space. We call this the Weighted Aggregation with Density Estimation module. We use this module times to group things together. Finally we test our method on two datasets: Oxford Robotcar and KITTI. Our method is better than methods by 7% to 8% on average.

B. The principles of neurodynamics, perceptrons and the theory of brain mechanisms are topics.

Part I of this study is about the background and basics of perceptrons. In Chapter 2 we look at ways to make brain models. Chapter 3 is about what makes a model of the brain. Chapter 4 has definitions and notation that we use later. Part II is three-layer series-coupled perceptrons. Part III is about -layer and cross-coupled perceptrons. Part IV is more speculative models and problems that we need to study more.

C. Gradient-based learning is a way to recognize documents.

Multilayer neural networks that use back-propagation are an example of this. Given the network architecture gradient-based learning can classify high-dimensional patterns like handwritten characters. This paper looks at methods for recognizing handwritten characters. We compare them on a task of recognizing handwritten digits. Convolutional neural networks are good at dealing with the variability of 2D shapes. Real-life document recognition systems have modules, including field extraction and language modeling. A new way of learning called graph transformer networks allows these systems to be trained globally using gradient-based methods.

D. ImageNet classification with convolutional neural networks is a challenging task.

We trained a deep convolutional neural network to classify 1.2 million high-resolution images into 1000 different classes. Our network has 60 million parameters and 650,000 neurons. We used -saturating neurons and a fast GPU implementation to make training faster. We also used a regularization method called dropout to prevent overfitting. Our network achieved top-1. Top-5 error rates of 37.5% and 17.0% on the test data. This is better than the state-of-the-art.

E. Neural machine translation is a way to do machine translation.

Unlike statistical machine translation neural machine translation uses a single neural network that can be tuned to maximize translation performance. The models we propose use an encoder-decoder architecture. The encoder encodes a source sentence into a fixed-length vector. The decoder generates a translation from this vector. We think that using a fixed-length vector is a limitation so we propose an approach that allows the model to automatically search for parts of the source sentence that are relevant to predicting a target word.

F. A decomposable attention model is a neural architecture for natural language inference.

We have come up with a neural architecture for natural language inference. This natural language inference approach uses attention to break down the problem into problems that can be solved one, by one which makes it very easy to work on these smaller problems at the same time. When we used this on the Stanford Natural Language Inference dataset we got the results so far with a lot fewer parameters than other people had used before and we did not need to know the order of the words. Natural language inference is what we are trying to improve. We also tried adding attention within sentences that considers a bit of word order and this made our natural language inference results even better.

G. BERT: Pre-training of bidirectional transformers for language understanding

Bert is a way to help computers understand language. It is called BERT, which means Bidirectional Encoder Representations from Transformers. This BERT thing is different from language models. BERT looks at both the words that come before and after a word to understand what it means. We can use BERT to make models that are really good at things like answering questions and understanding language. We do not have to change a lot to make these models. BERT is an idea but it works really well. BERT is really good at a lot of language tasks. It got the results ever on eleven tasks. For example it got 80.5 percent on the GLUE test, which's 7.7 percent better than before. It also got 86.7 percent on the MultiNLI test, which's 4.6 percent better. BERT is also good at answering questions. It got 93.2 percent on the SQuAD v1.1 test and 83.1 percent on the SQuAD v2.0 test. BERT is really good at understanding language. The BERT model can be used to make a lot of models that are good, at language tasks.

III. SYSTEM ARCHITECTURE

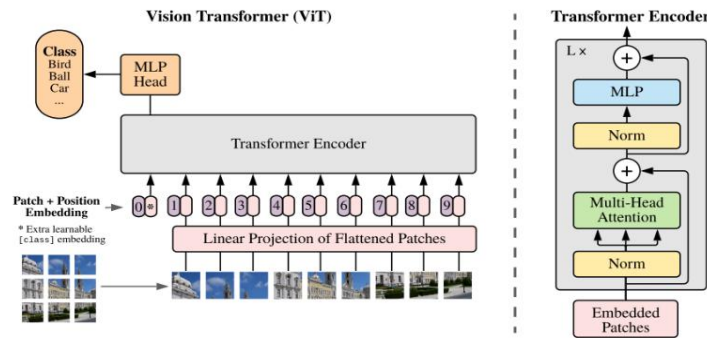


Fig:1 System Architecture of Vision Transformer (ViT)

The picture shows how a Vision Transformer (ViT) model works for recognizing things in images. First the image is broken down into pieces called patches. These patches are then made flat. Sent through a special layer that turns them into a list of patch embeddings. The Vision Transformer (ViT) model also adds some information to these embeddings so it can remember where each patch is in the image. It adds a token that represents the whole image too.

These embeddings are then sent to a part of the Vision Transformer (ViT) model called the Transformer Encoder. This encoder has layers stacked on top of each other. Each layer looks at how all the patches are related to each other which helps the Vision Transformer (ViT) model understand the image better. After that it uses some steps to make sure it learns from the patches. The Vision Transformer (ViT) model keeps looking at the patches over to learn more about the image.

Finally the Vision Transformer (ViT) model uses the information it learned to figure out what is in the image, like a bird or a car. It does this by using a part called the MLP head. This architecture is special because it uses a way of looking at the image instead of the usual way that convolutional operations work. The Vision Transformer (ViT) model looks at the image and how all the parts are related, which helps it recognize things more accurately.

IV. METHODOLOGY

TABLE I: Summary of Methodology Components and Functional Description

No.	Module	Functional Description
1	Model Loading Module	Loads Vision Transformer (ViT) and YOLO models for image classification and detection
2	Image Upload Module	Allows the user to upload an input image for processing
3	Image Classification using ViT	Classifies the uploaded image into categories using Vision Transformer
4	Object Detection using YOLO	Detects and identifies objects present in the image using YOLO
5	Accuracy Calculation Module	Calculates model accuracy based on prediction results
6	Comparison Graph Module	Displays a graph comparing model performance
7	User Interface Module	Provides interface for user interaction and system control
8	Result Display Module	Displays classification and detection results to the user

Table 1 describes our proposed Vision Transformer system. This system has parts, for classifying images and detecting objects. It begins with two modules: one loads the model and the other uploads images. Then it uses ViT to classify images and YOLO to detect objects. The system also has a module that calculates how accurate it is and another module that shows the results in a graph. The User Interface helps users interact with the system. The Result Display Module then clearly shows what the system found.

A. Model Loading Module

The Vision Transformer-based image classification and object detection system starts by loading the machine learning models that it needs. The Vision Transformer model and the YOLO model are loaded into the computers memory. These models have already been trained on a lot of data so they know a lot about objects and what images look like.

The Vision Transformer model is mainly used to classify images in detail. The YOLO model is used to find objects. When the models are loaded some important settings are also set up like the size of the images and what the models are looking for. This step is very important because if the models are not loaded correctly the Vision Transformer-based image classification and object detection system cannot do its job.

B. Image Upload Module

In this part the user gives the Vision Transformer-based image classification and object detection system an image to work with. The user can choose an image from their computer. Upload it to the Vision Transformer-based image classification and object detection system. The image is then shown on the screen.

This step makes sure that the user can see the image before the Vision Transformer-based image classification and object detection system starts working on it. The image is also made to fit the needs of the Vision Transformer model and the YOLO model. This is important so that the Vision Transformer-based image classification and object detection system can do a job.

C. Image Classification using VIT

After the image is uploaded it is sent to the Vision Transformer model to be classified. The Vision Transformer model looks at the image in parts and uses a special way of paying attention to understand what is in the image.

The Vision Transformer model is very good at classifying images in detail. For example it can say that an object is not a dog but a Labrador or a German Shepherd. The Vision Transformer model looks at the image makes a guess and figures out how sure it is.

The Vision Transformer-based image classification and object detection system then calculates how accurate the guess is. The result is shown on the image with the name of what the Vision Transformer model thinks it is. This part of the Vision Transformer-based image classification and object detection system is very important because it helps us understand what is in the image.

D. Object Detection using YOLO

This part of the Vision Transformer-based image classification and object detection system uses the YOLO model to find objects in the image. The YOLO model is very fast. Can look at the whole image at once.

The YOLO model finds objects in the image. Draws boxes around them. Each box has the name of what the YOLO model thinks the object is and how sure it is. For example if there is a dog and a car in the image the YOLO model will find both. Highlight them.

The YOLO model is very good at finding objects. It does not classify them in detail like the Vision Transformer model. This makes it very useful for finding objects in time. The results are shown in a way that's easy to understand.

E. Accuracy Calculation Module

After the Vision Transformer model and the YOLO model are done the Vision Transformer-based image classification and object detection system calculates how well they did. In this part the accuracy of both models is calculated.

For the Vision Transformer model accuracy is calculated by looking at how sure it was. For the YOLO model accuracy is calculated by looking at how it found objects. These numbers are then averaged to get a score.

This part is important because it helps us compare how well the Vision Transformer model and the YOLO model did. By looking at the accuracy users can see which model is better in situations.

F. Result Display Module

The final part of the Vision Transformer-based image classification and object detection system shows the user the results. It displays the image with what the Vision Transformer model thinks it is and what objects the YOLO model found.

The result is shown in a way that's easy to understand. This part is very important because it helps the user see what the Vision Transformer-based image classification and object detection system did.

V. EXPERIMENTAL RESULTS AND EVALUATION

A. Dataset Description

We worked with a set of images that had labels on them to do some experiments with classification and object detection. The Vision Transformer model uses a set of images called ImageNet that has more than 1 million pictures in 1000 different groups. The YOLO model was trained on the COCO set which has 80 kinds of things like animals and cars and things we see every day.

The images we used to test our work had lots of real life pictures like dogs and cats and other things. We used these pictures to see how well both models worked. Since we used sets of images that were already prepared we did not have to make sure the images were balanced. The models were already good at handling different kinds of images.

B. Experimental Setup

The images used in this project are given to the system by the user when it is running and these images are worked on using two models: the Vision Transformer model and the YOLO model. First the images need to be prepared before they can be used. This means they have to be made smaller to 384 by 384 pixels for the Vision Transformer model and they have to be changed into the format.

The YOLO model on the hand uses OpenCV to work on the images directly. The system does things in an order. It loads the Vision Transformer and YOLO models that have already been trained. Then it uploads the image that the user wants to use. The Vision Transformer model is used to figure out what is in the image. The YOLO model is used to find objects in the image. The system then calculates how well both models are doing.

Finally it shows a graph that compares how well the Vision Transformer model and the YOLO model are doing. We tried out the system on a computer, with an Intel i3 processor 4GB of memory and a Windows operating system. We did not use any graphics card and the system still worked well. This shows that the system can work properly on computers that are not very powerful and it does not need any special hardware. The Vision Transformer model and the YOLO model are used together in the system to make sure it works well.

C. System Interface

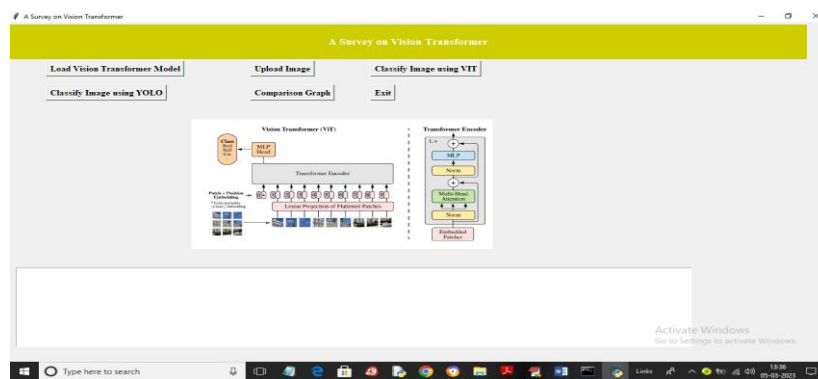


Figure 2. Screenshot of Vision Transformer system. It shows what the main screen looks like. The Vision Transformer system has a lot of options. These options are Load Vision Transformer Model, Upload Image Classify Image using Vision Transformer Classify Image using YOLO and Comparison Graph. The Vision Transformer system also has some features that are highlighted.

Figure 2 is the screen of the Vision Transformer application. This screen has a lot of options. You can Load Vision Transformer Model. You can Upload Image. The Vision Transformer application also lets you Classify Image using VIT and Classify Image using YOLO. There is a Comparison Graph option.. If you are done you can Exit.

The screen also shows a diagram of the Vision Transformer model. This helps people understand how the Vision Transformer model works with images. The Vision Transformer application is easy to use. People can use the Vision Transformer application to classify images and detect objects easily. The Vision Transformer application is very user-friendly.

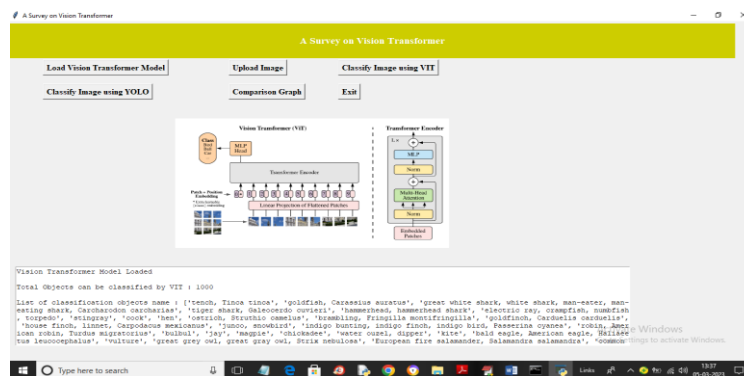


Figure 2. This is a picture of the Vision Transformer system. It shows how to load a model upload an image and use Vision Transformer and YOLO for classification. You can also see options, for comparison graphs.

We are testing a system that uses deep learning models to look at pictures and find things in them. This system takes pictures from the user. Looks at them using two models: the Vision Transformer and the YOLO model. The system has an interface where users can load models upload pictures look at what is in the pictures using Vision Transformer find things in the pictures using YOLO and look at comparison graphs.

The Vision Transformer model is used to say what is in the picture by looking at what the picture's about while the YOLO model is used to find things in the picture and draw boxes around them. The system also shows how things it can find in the picture and shows what kinds of things the model can look for. The interface has all the options you need to do these things

The system works on a computer without a graphics card, which shows that it can work well on computers that are not very powerful. The results show that the Vision Transformer is very good at saying what is in the picture while YOLO is very good at finding things in the picture accurately. This shows that using both models makes the system work better.

In brief this project is about a system that can look at pictures and find things in them using computer techniques. It uses Vision Transformer to say what is in the picture and YOLO to find things in the picture. The system is easy to use and gives answers, which makes it useful for real things like looking at pictures and making smart machines.

The system uses Vision Transformer and YOLO models to make it work. Vision Transformer and YOLO models are very important for the system. The system is, about using Vision Transformer and YOLO models to look at pictures.

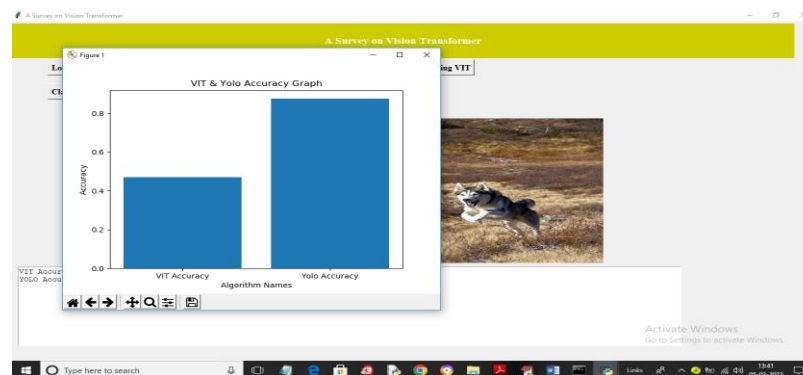


Figure 3. This is a graph that compares how accurate the VIT model and the YOLO model are. It also shows the picture that was used as input, for the VIT model and the YOLO model.

We are trying out a system that sees how well two deep learning models work: the Vision Transformer and the YOLO model. This system takes a picture. Uses both models to look at it. The Vision Transformer says what is in the picture and the YOLO model finds the things in the picture.

The results we get include a bar graph that compares how well both models do their job. The graph shows that YOLO is better at finding things than the Vision Transformer. The Vision Transformer is better at saying exactly what the things are. We also get to see the picture that the system is looking at.

This comparison graph helps us understand how both models work on the picture. It shows us what each model is good at: YOLO is good at finding things and the Vision Transformer is good at saying what the things are.

So this system looks at two models. Compares them using a graph, which makes it easier to see how well they work and what they are good at when it comes to looking at pictures and finding things in them. The Vision Transformer and the YOLO model are both useful, in ways.

VI. FUTURE SCOPE AND CONCLUSION

The Vision Transformer has shown a lot of potential in computer vision tasks. However there are areas where the Vision Transformer can be improved. One important area to focus on is making the Vision Transformer more efficient. The Vision Transformer requires a lot of power and large datasets to work properly.

Future research can focus on creating a more optimized Vision Transformer that can run smoothly on devices like mobile phones and embedded systems.

Another area for improvement is combining the Vision Transformer with Convolutional Neural Networks. This combination can help improve feature extraction and reduce the amount of computations needed. The system can also be improved to support real-time video processing and tracking objects at the same time. This will make the Vision Transformer more suitable for applications like surveillance and autonomous systems.

More research is needed to improve the accuracy of the Vision Transformer. This can be done by tuning the model using transfer learning and using larger and more diverse datasets. Adding modalities like text and audio can also improve the performance of the system in complex real-world applications. The future of the Vision Transformer is promising and ongoing advancements will make it more practical and efficient for a wide range of applications.

In this paper we talked about the Vision Transformer and its application in image classification along with YOLO for object detection. Our study shows that the Vision Transformer is an alternative to traditional convolutional neural networks in computer vision tasks. The Vision Transformer can classify images into categories and YOLO can detect objects quickly and accurately.

Our experimental results show that both models work efficiently and complement each other. YOLO is better at detecting objects while the Vision Transformer is better at classifying objects into grained categories. The comparison graph clearly shows the strengths and weaknesses of both approaches. The system is designed to be user-friendly. Can work smoothly even on devices with limited resources without needing a GPU.

Overall our proposed system shows that combining the Vision Transformer and YOLO improves performance and provides results for real-world applications. This work highlights the growing importance of transformer-based models in computer vision. Opens up new possibilities for future research and development, in intelligent image processing systems.

VII. ACKNOWLEDGMENT

The authors would like to thank the Department of Computer Science and Engineering, CMR Technical Campus for providing support and guidance for this project. Special thanks to our project guide for continuous encouragement.

VIII. REFERENCES

- [1] F. Rosenblatt. The perceptron, a perceiving and recognizing automaton Project Para. Cornell Aeronautical Laboratory, 1957.
- [2] F. ROSENBLATT. Principles of neurodynamics. perceptrons and the theory of brain mechanisms. Technical report, 1961.
- [3] Y. LeCun et al. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.
- [4] A. Krizhevsky et al. Imagenet classification with deep convolutional neural networks. In *NeurIPS*, pp. 1097–1105, 2012.
- [5] D. E. Rumelhart et al. Learning internal representations by error propagation. Technical report, 1985.
- [6] S. Hochreiter and J. Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997.
- [7] D. Bahdanau et al. Neural machine translation by jointly learning to align and translate. In *ICLR*, 2015.
- [8] A. Parikh et al. A decomposable attention model for natural language inference. In *EMNLP*, 2016.
- [9] A. Vaswani et al. Attention is all you need. In *NeurIPS*, 2017.
- [10] J. Devlin et al. Bert: Pre-training of deep bidirectional transformers for language understanding. In *NAACL-HLT*, 2019. [11] T. B. Brown et al. Language models are few-shot learners. In *NeurIPS*, 2020.
- [12] K. He et al. Deep residual learning for image recognition. In *CVPR*, pp. 770–778, 2016.
- [13] S. Ren et al. Faster R-CNN: Towards real-time object detection with region proposal networks. In *NeurIPS*, 2015.
- [14] M. Chen et al. Generative pretraining from pixels. In *ICML*, 2020.
- [15] A. Dosovitskiy et al. An image is worth 16x16 words: Transformers for image recognition at scale. In *ICLR*, 2021.
- [16] N. Carion et al. End-to-end object detection with transformers. In *ECCV*, 2020.
- [17] X. Zhu et al. Deformable detr: Deformable transformers for end-to-end object detection. In *ICLR*, 2021.
- [18] S. Zheng et al. Rethinking semantic segmentation from a sequence-to sequence perspective with transformers. In *CVPR*, 2021.
- [19] H. Chen et al. Pre-trained image processing transformer. In *CVPR*, 2021.
- [20] L. Zhou et al. End-to-end dense video captioning with masked transformer. In *CVPR*, pp. 8739–8748, 2018.
- [21] S. Ullman et al. High-level vision: Object recognition and visual cognition, volume 2. MIT press Cambridge, MA, 1996.
- [22] R. Kimchi et al. Perceptual organization in vision: Behavioral and neural perspectives. Psychology Press, 2003.
- [23] J. Zhu et al. Top-down saliency detection via contextual pooling. *Journal of Signal Processing Systems*, 74(1):33–46, 2014.
- [24] J. Long et al. Fully convolutional networks for semantic segmentation. In *CVPR*, 2015.
- [25] H. Wang et al. Max-deeplab: End-to-end panoptic segmentation with mask transformers. In *CVPR*, pp. 5463–5474, 2021.
- [26] Robert B. Fisher has a compendium of computer vision. This was updated in 2008.
- [27] Niki Parmar and others worked on the image transformer. They presented this at *ICML* in 2018.
- [28] Yujin Zeng and others learned spatial-temporal transformations for video inpainting. They presented this at *ECCV* in 2020.
- [29] Kai Han and others worked on Transformer in Transformer. They presented this at *NeurIPS* in 2021.
- [30] Ze Liu and others worked on Swin-Unet, which's a pure transformer for medical image segmentation. They published this in 2021.
- [31] Xiaokang Chen and others did a study of training self-supervised vision transformers. They presented this at *ICCV* in 2021.

- [32] Zhaohui Dai and others worked on UP-DETR, which's for unsupervised pre-training for object detection with transformers. They presented this at CVPR in 2021.
- [33] Yuxin Wang and others did end-to-end video instance segmentation with transformers. They presented this at CVPR in 2021.
- [34] Liu Huang and others worked on Hand-Transformer, which's for 3D hand pose estimation. They presented this at ECCV in 2020.
- [35] Liu Huang and others worked on HOT-Net, which's for 3D hand-object pose estimation. They presented this at ACM MM in 2020.

Copyright & License:

© Authors retain the copyright of this article. This work is published under the Creative Commons Attribution 4.0 International License (CC BY 4.0), permitting unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.