Alma Mater Studiorum - University of Bologna

Department of Computer Science and Engineering

Master's Degree in Artificial Intelligence

Final Thesis in
Natural Language Processing

SynBA: A contextualized Synonim-Based adversarial Attack for Text Classification

Supervisor
Prof. Paolo Torroni

Co-supervisors
Dr. Federico Ruggeri
Dr. Giulia De Poli

Candidate
Giuseppe Murro

Academic Year 2021-2022- Third session

Dedica

Abstract

This dissertation describes a deepening study about Visual Odometry problem tackled with transformer architectures. The existing VO algorithms are based on heavily hand-crafted features and are not able to generalize well to new environments. To train them, we need carefully fine-tune the hyper-parameters and the network architecture. We propose to tackle the VO problem with transformer because it is a general-purpose architecture and because it was designed to transformer sequences of data from a domain to another one, which is the case of the VO problem.

Our first goal is to create synthetic dataset using BlenderProc2 framework to mitigate the problem of the dataset scarcity. The second goal is to tackle the VO problem by using different versions of the transformer architecture, which will be pre-trained on the synthetic dataset and fine-tuned on the real dataset, KITTI dataset.

Our approach is defined as follows: we use a feature-extractor to extract features embeddings from a sequence of images, then we feed this sequence of embeddings to the transformer architecture, finally, an MLP is used to predict the sequence of camera poses.

“Happiness can be found even in the darkest of times, when one only remembers to turn on the light.”
– Dumbledore

Thanks

First, I would like to express my deepest gratitude to Professor Luigi Di Stefano and Luca De Luigi for the guidance and support during the internship

Second, I would like to thank my parents for their moral support and for their patience.

Third, I would like to thank my girlfriend and friends for being patient with me and to put up with my complainings.

Bologna, 06 December 2022 Giuseppe Murro

Contents

List of Figures

List of Tables

Chapter 1

Introduction

In this section we will present the summarized content of the whole thesis.

The introduction allows you to orient the reader to your research project and preview the organisation of your thesis. In the introduction, state what the topic is about, explain why it needs to be further researched and introduce your research question(s) or hypothesis.

Whilst patterns of organisation in introductions vary, there are some common features that will help you to achieve an informative and engaging introduction. Let' s identify these features:

Introduce the topic Define key terms and concepts Give background and context for the topic (this may include a brief literature review) Review and evaluate the current state of knowledge in the topic (this may include a brief literature review) Identify any gaps, shortcomings and problems in the research to date Introduce your research question(s) or hypothesis Briefly describe your methodology and/or theoretical approach Explain the aim of your research and what contribution it will make to the topic Give an overview of the chapter outline of the thesis. It' s important to note that, depending on your field of study and the faculty requirements of your thesis, not all of these features will be relevant. Also, these features may occur in varied orders.

Most people write many drafts of their introduction. It can be useful to write one early in the research process to clarify your thinking. You will need to write a version for your confirmation proposal and other milestones. As your research progresses and your ideas develop, you will need to revise it. When the final draft of chapters is complete, check the introduction once more to make sure that it accurately reflects what you have actually done.

1.1 Topic definition

1.2 Problem statement

1.3 Research question

1.4 Solution

We tried to tackle the problem by designing a deep neural network which is composed by a feature extractor, the transformer and a MLP to predict the pose. We feed the feature extractor with a sequence of images, we tried both grey-scale and RGB images, in this way, we obtain a sequence of embeddings (both size 512 and 2048), the embedding are then fed into the transformer (both encoder-only and encoder-decoder version) and the output of the transformer is fed into the MLP to predict the sequence of poses.

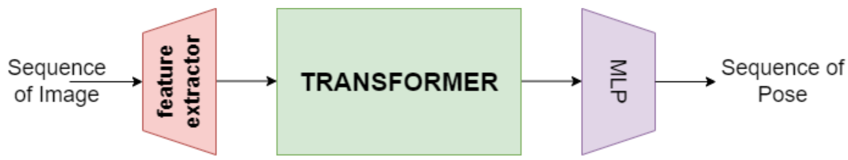

Figure 1.1: General representation of the model.

We use a sequence of image because the transformer model, originally designed for the machine translation, it requires as input a sequence of embeddings, then it outputs another sequence of embeddings. For major details about the transformer, we refer to §4.1 Experiments - Models and §5.3 Implementation - Models.

1.5 Thesis organization

First chapter introduces the general content about thesis and gives a short presentation of the topic, the problem and the solution we propose;

Second chapter a deepening about the theoretical foundations used during the stage and the project;

Third chapter presents the datasets used during for the training and the testing of the model;

Fourth chapter presents the experiments did during to develop the system;

Fifth chapter discusses about the results and possible future developments.

During the drafting of the essay, following typography conventions are considered:

- the acronyms, abbreviations, ambiguous terms or terms not in common use are defined in the glossary, in the end of the present document;
- the first occurrences of the terms in the glossary are highlighted like this: word;
- the terms from the foreign language or jargon are highlighted like this: *italics*.

Chapter 2

Background

In this chapter we will present the theoretical knowledge useful to understand the content from successive chapters.

2.1 Natural Language Processing

Deep learning method is part of machine learning methods based on artificial neural network with representation learning. The learning process can be supervised, semi-supervised, or unsupervised.

There is a very large variety of deep learning architectures, some of them are specialized in some fields meanwhile others have a broader usage, especially, there are CNNs and Transformers.

In recent years, the field of computer vision has been growing in complexity and the number of applications has been increasing, in addition to those presented in Section 1.1 Computer vision, there are Simultaneous Localization and Mapping (SLAM) and visual odometry which is a task in which the robot is able to understand where it is and how it is oriented.

The development of computer vision has been a long process, the growth is favoured by the development of new hardware components and new challenges, about the latters, we have CIFAR-10 (Doon et al. [3]), Fashion-MNIST(Xiao et al. [22]), MS-Coco (Lin et al. [11]) and ImageNet (Deng et al. [2]). These datasets are often used as benchmark for novel approaches.

For the architectures, starting from AlexNet (Krizhevsky et al. [10]), then VGG (Simonyan et al. [15]), Inception-V1 (Szegedy et al. [16]), Inception-V2 (Szegedy et al. [17]), ResNet (He et al. [7]), etc., the complexity of the models has increased enormously. Each of these models introduced some innovations and improved the performance on the benchmarks, for example:

- AlexNet introduced the concept of the *convolutional neural network* (CNN) and use of the separation of the models into two different GPUs.
- VGG introduced the concept of stage, which repeated more times, composes the model.
- Inception-V1, Inception-V2 and Inception-V3 which are based on the concept of *inception module* which was composed by different paths that the input has to go through to reach the output.

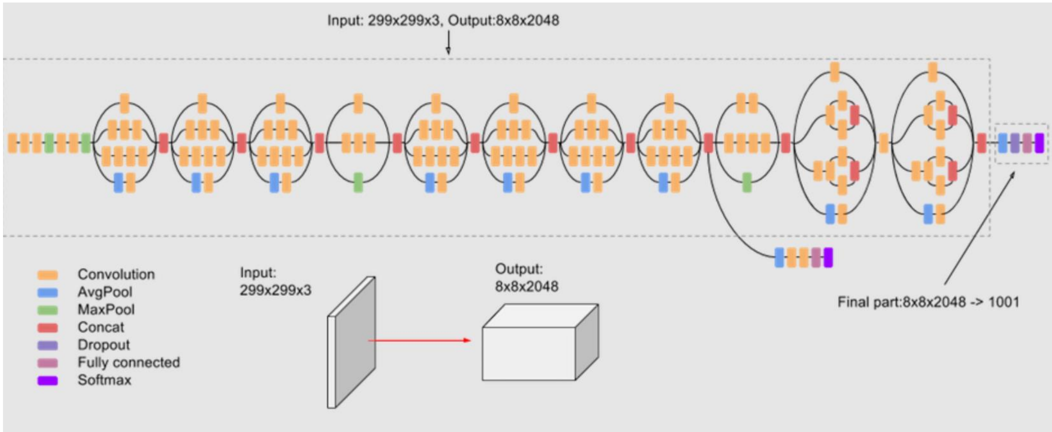

Figure 2.1: Inception V3 Structure.

- ResNet is a model that is based on the concept of *residual network* which is composed by several blocks of the same type with the skip connections:

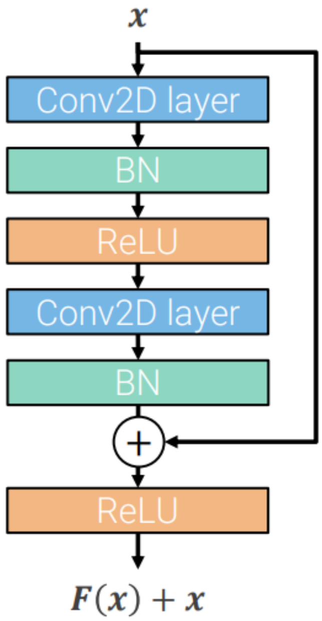

Figure 2.2: Skip connection.

> Basically, the input of the block is added to the output before feeding it to the next block, in this way, we can avoid the vanishing gradient problem making easier the training process.

After this, the computer-visionists lend the Transformer architecture (Vaswani et al. [19]) from Natural Language Processing (NLP), bringing up ViT (Dosovitskiy et al. [4]) which is based on the Multi-Head Attention (MHA) mechanism. A multi-head attention is a module of attention mechanisms repeated several times in parallel. In this way, the model can attend to different parts of the input, forming the cross-attention over different parts of the input. For major details, please refer to §2.1.2 Transformer.

2.1.1 Text classification

The Convolutional Neural Network (CNN) is a class of artificial neural network, it is used in almost every imagery related task, such as image classification, object detection, image segmentation, etc.

The CNN takes an input image, assign importance (learnable weights and biases) and process the input image by using the convolution operation extracting features. There are two important parameters in the convolution operation, the kernel size and the stride. The kernel is a matrix which is used to perform the convolution operation, the stride is the number of pixels the kernel slides each step over the input image to produce a new pixel of the output feature map. With stride, we can control the size of the output image, if the stride is equal to 1, the output image will have the same size of the input image, if the stride is equal to 2, the output image will have half the size of the input image.

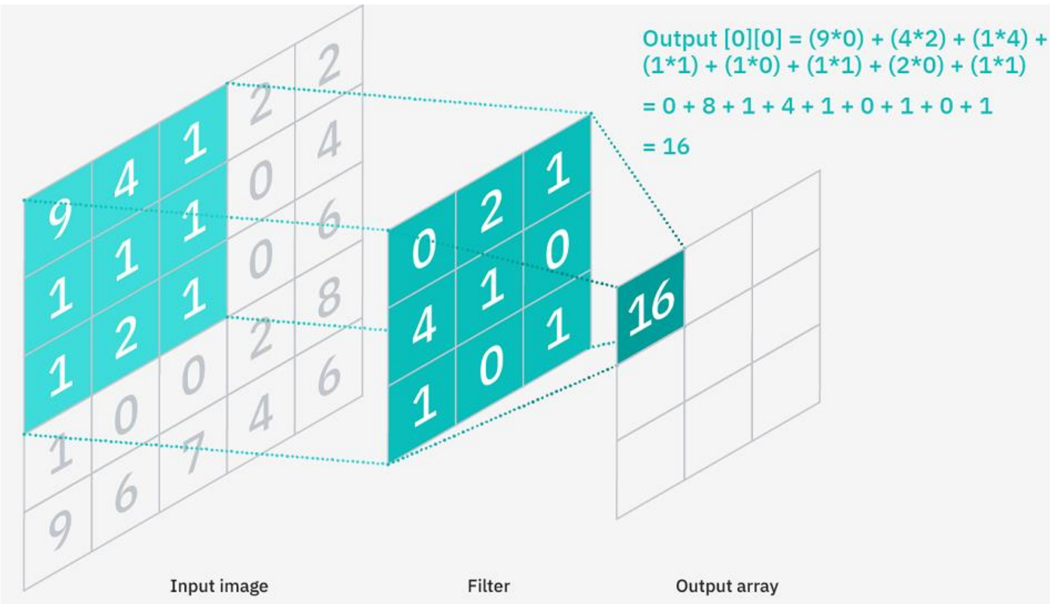

Figure 2.3: Convolutions: every single element of the output feature map is obtained by summing the element-wise product between the elements from the input feature map and the kernel. The whole feature map is then obtained sliding the kernel over the input feature map.

Then, there are pooling layers, usually max-pooling and average pooling, which can reduce the dimensionality of the feature maps by setting strides $>= 2$, which is useful to reduce the computational cost. An important property of max-pooling is that it is translation invariant, which means that the output of the max-pooling layer is the same regardless of the position of the input feature map. For example, max-pooling is computed as showed in the image:

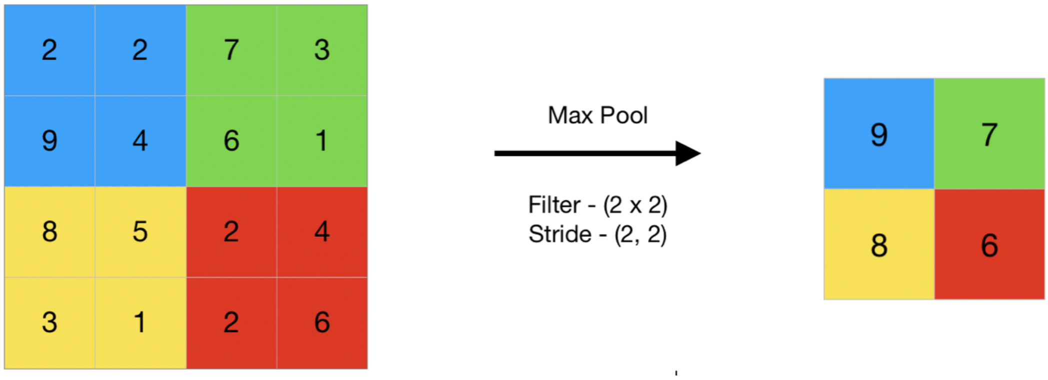

Figure 2.4: Max-pooling: essentially, it strides over the input image and takes the max value of the area covered by the kernel.

With stride = 2, sliding over the input feature map and taking the maximum value of the window, the dimensionality of the feature map is reduced.

Another important component is the activation function, Rectified Linear Unit (ReLU) is the most used one, it is defined as:

$$ReLU(x) = \max(0, x) \tag{2.1}$$

It guarantees the non-linearity of the network, allowing the network to learn more complex features. These are the main components of a CNN, but there are other components, such as batch normalization and dropout which are used to improve the performance of the network reducing the over-fitting. Increasing the number of layers and combining the pooling layers, the CNN is able to extract more and more complex features, such as edges, lines, shapes, etc. Currently, the most used CNN architecture is the ResNet which will be used in the project as feature-extractor.

2.1.2 Sentiment analysis

The transformer architecture is a class of neural network architecture, born for the task of machine translation, but it has been used in many vision tasks.

As introduced in [19], the Transformer is a model architecture based entirely on attention mechanism. Which can be divided into different steps: the first step of attention mechanism is to compute the Q, K and V vectors, by multiplying the input vector x by the weight matrices W_q, W_k and W_v. Then, the attention weights are computed by using the scaled dot product attention, which is the softmax of the dot product between the query and the key vectors divided by the square root of the dimensionality of the key vector. Finally, the attention weights are multiplied by the value vector to obtain the output vector. The output vector is then passed through a feed-forward neural network, which is composed by two linear layers with ReLU activation function, added to the input vector and normalized by the layer normalization. The self-attention module is then repeated N times, where N is the number of layers.

The whole process can be summarized as the:

$$Attention(Q, K, V) = softmax\left(\frac{QK^T}{\sqrt{d_k}}\right)V \tag{2.2}$$

And the graphical representation is:

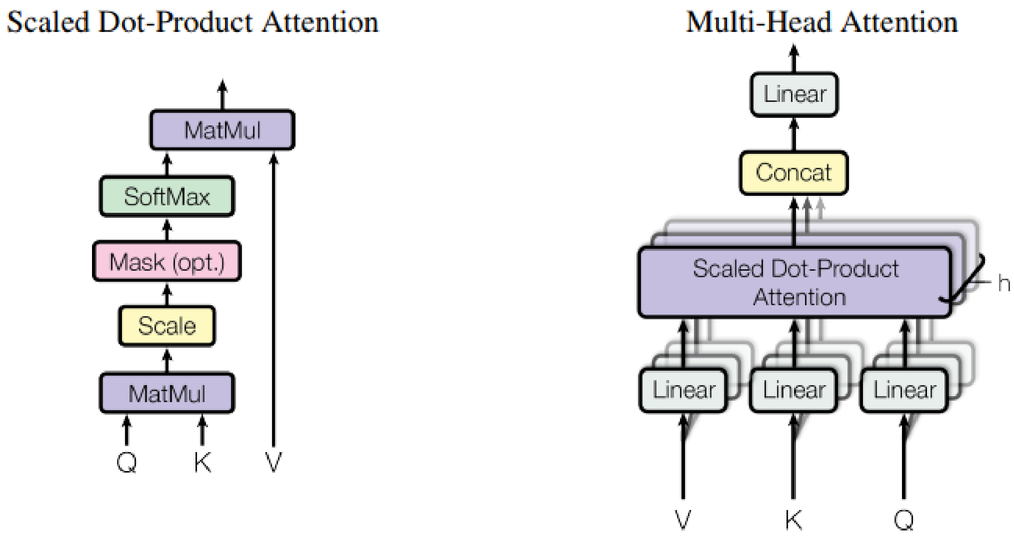

Figure 2.5: Attention mechanism: (Left) scaled-dot-product attention. (right) Multi-head attention which is obtained by combining many scaled-dot-product attention.

An important notion introduced is the multi-head attention, which is using more scaled dot product attention, each one with different weights, and concatenating each output vectors. And this is the transformer encoder.

Then, using a slightly modified version of the self-attention module, we obtain the decoder, which takes as input also the output sequence from encoder, and repeating the number of encoder and decoder modules, we obtain the whole transformer architecture, as follows:

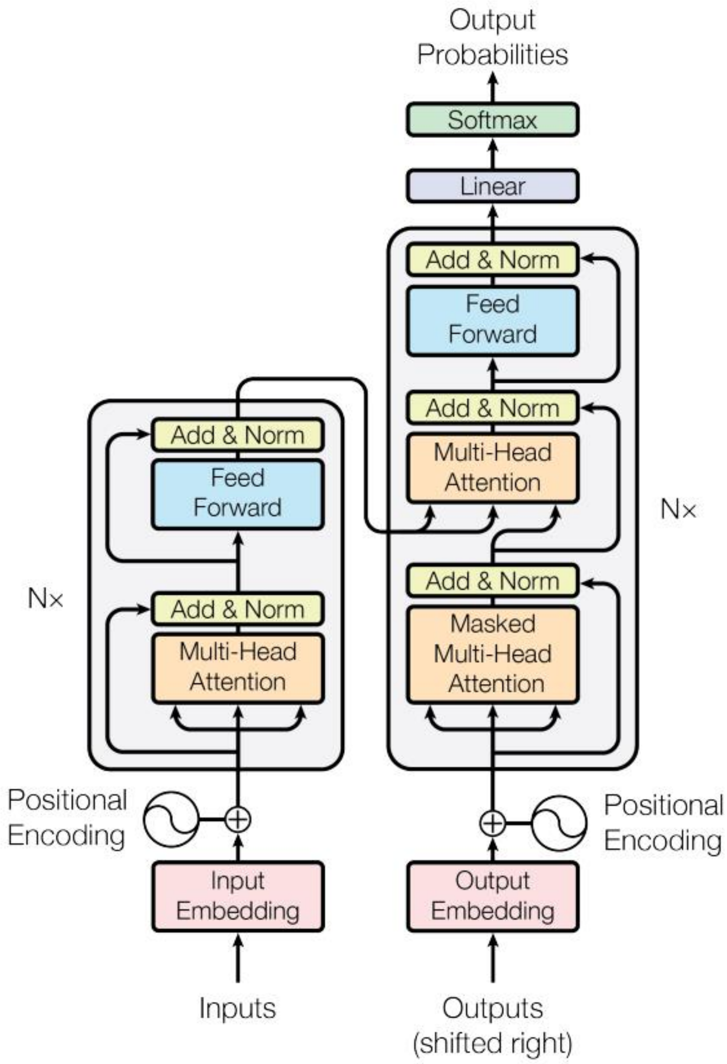

Figure 2.6: Transformer architecture: the encoder and decoder modules are composed by a self-attention module and a feed-forward neural network repeated N times.

With this architecture the new state-of-the-art results have been achieved in Natural Language Processing, especially in machine translation (a seq2seq task). Then the adapted version applied to vision tasks also brought very good results, such as in object detection, image captioning, etc.

Another important notion is the mask which is used in sequence-to-sequence models, such as the transformer more specifically in the decoder, to avoid the model to attend to the future tokens. To ensure this, the future positions are masked with $-\infty$ before the softmax step in the self-attention calculation.

2.1.3 Natural language inference

2.1.4 Sequence-to-Sequence models

2.1.5 Lexicon

2.1.6 Word Embeddings

2.1.7 Masked Language Models

2.2 Adversarial Machine Learning

2.2.1 Adversarial attacks

2.2.2 Paradigm shift: from Computer Vision (CV) to NLP

2.2.3 Taxonomy of textual adversarial attacks

2.2.4 Adversarial attack methods from literature

2.2.4.1 TextFooler

2.2.4.2 BERT-based attacks

2.3 Machine Learning hardening

2.3.1 Vanilla adversarial training

2.3.2 Attacking to Training

2.4 Text Attack

2.4.1 Framework structure

2.4.2 Attack components

2.4.3 HuggingFace integration

Chapter 3

Methodology

In this chapter we will present the datasets created and used for the visual odometry.

- Introduction
- Research Design
- Research Questions and Hypotheses
- Setting and Sample
- Data Collection
- Data Analysis
- Conclusion

demonstration of fit between methods chosen and research question(s) rationale for choosing materials, methods and procedures details of materials, equipment and procedures that will allow others to: replicate experiments understand and implement technical solutions

3.1 Defined goal

3.1.1 Problem to solve

3.1.2 Research objective

3.2 Research design

3.2.1 Models to attack

3.3 Proposed solution

3.3.1 Intuition

3.3.2 SynBA components

3.3.2.1 Search Method

3.3.2.2 Transformation

3.3.2.3 Constraints

3.3.2.4 Goal Function

3.3.3 Hyperparameter Tuning

3.3.4 Candidates ranking calibration

3.4 Evaluation metrics

3.4.1 Attack metrics

3.4.2 Quality metrics

3.4.3 Performance metrics

Chapter 4

Experimental results

In this chapter we will discuss about different models and different prediction strategies.

4.1 Data collection

4.1.1 Experimental setup

4.1.2 Datasets perturbed

4.1.3 Model attacked

4.2 Qualitative evaluation

4.2.1 Results on rotten-tomatoes

4.2.2 Results on imdb

4.3 Quantitative evaluation

4.3.1 Performances on rotten-tomatoes

4.3.2 Performances on imdb

4.4 Human evaluation

Chapter 5

Final discussions

In this chapter we will discuss the results achieved, future developments and personal comments.

A clear answer to your research question or hypothesis Summary of the main findings or argument Connections between your findings or argument to other research Explanation and significance of the findings Implications of the findings Limitations of the research and methodology Recommendations for future research

Summary of Findings Limitations of the research Suggestions for Future Research Conclusion

5.1 Summary of findings

5.2 Limitations

5.3 Future developments

5.4 Conclusions

Bibliopraphy

Paper references

[1] Alkendi. "State of the Art in Vision-Based Localization Techniques for Autonomous Navigation Systems". In: *IEEE Access* PP (May 2021), pp. 1–1. DOI: 10.1109/ACCESS.2021.3082778.

[2] Jia Deng et al. "ImageNet: A large-scale hierarchical image database". In: (2009), pp. 248–255. DOI: 10.1109/CVPR.2009.5206848 (cit. on p. 5).

[3] Raveen Doon, Tarun Kumar Rawat, and Shweta Gautam. "Cifar-10 Classification using Deep Convolutional Neural Network". In: (2018), pp. 1–5. DOI: 10.1109/PUNECON.2018.8745428 (cit. on p. 5).

[4] Alexey Dosovitskiy et al. "An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale". In: (2020). DOI: 10.48550/ARXIV.2010.11929. URL: https://arxiv.org/abs/2010.11929 (cit. on p. 7).

[5] "Euler angles". In: (). URL: https://mathworld.wolfram.com/EulerAngles.html.

[6] Andreas Geiger, Philip Lenz, and Raquel Urtasun. "Are we ready for autonomous driving? The KITTI vision benchmark suite". In: (June 2012), pp. 3354–3361. ISSN: 1063-6919. DOI: 10.1109/CVPR.2012.6248074.

[7] Kaiming He et al. "Deep Residual Learning for Image Recognition". In: (2015). DOI: 10.48550/ARXIV.1512.03385. URL: https://arxiv.org/abs/1512.03385 (cit. on p. 5).

[8] "Hebbian Plasticity". In: (1949). URL: https://en.wikipedia.org/wiki/Hebbian_theory.

[9] Berthold Horn. “Closed-Form Solution of Absolute Orientation Using Unit Quaternions”. In: *Journal of the Optical Society A* 4 (Apr. 1987), pp. 629–642. DOI: 10.1364/JOSAA.4.000629.

[10] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. “ImageNet Classification with Deep Convolutional Neural Networks”. In: 25 (2012). URL: https://proceedings.neurips.cc/paper/2012/file/c399862d3b9d6b76c8436e924a68c45b-Paper.pdf (cit. on p. 5).

[11] Tsung-Yi Lin et al. “Microsoft COCO: Common Objects in Context”. In: (2014). DOI: 10.48550/ARXIV.1405.0312. URL: https://arxiv.org/abs/1405.0312 (cit. on p. 5).

[12] David Prokhorov et al. “Measuring robustness of Visual SLAM”. In: (2019). DOI: 10.48550/ARXIV.1910.04755. URL: https://arxiv.org/abs/1910.04755.

[13] Joseph Redmon and Ali Farhadi. “YOLOv3: An Incremental Improvement”. In: (2018). DOI: 10.48550/ARXIV.1804.02767. URL: https://arxiv.org/abs/1804.02767.

[14] Stephan R. Richter, Zeeshan Hayder, and Vladlen Koltun. “Playing for Benchmarks”. In: (2017). DOI: 10.48550/ARXIV.1709.07322. URL: https://arxiv.org/abs/1709.07322.

[15] Karen Simonyan and Andrew Zisserman. “Very Deep Convolutional Networks for Large-Scale Image Recognition”. In: (2014). URL: https://arxiv.org/abs/1409.1556 (cit. on p. 5).

[16] Christian Szegedy et al. “Going Deeper with Convolutions”. In: (2014). URL: https://arxiv.org/abs/1409.4842 (cit. on p. 5).

[17] Christian Szegedy et al. “Rethinking the Inception Architecture for Computer Vision”. In: (2015). URL: https://arxiv.org/abs/1512.00567 (cit. on p. 5).

[18] “Transformer in Computer Vision”. In: (). URL: https://www.axelera.ai/vision-transformers-in-computer-vision/.

[19] Ashish Vaswani et al. “Attention Is All You Need”. In: (2017). DOI: 10.48550/ARXIV.1706.03762. URL: https://arxiv.org/abs/1706.03762 (cit. on pp. 7, 9).

[20] Hongjian Wang et al. "Monocular VO Based on Deep Siamese Convolutional Neural Network". In: *Complexity* 2020 (Mar. 2020), pp. 1–13. DOI: 10.1155/2020/6367273.

[22] Han Xiao, Kashif Rasul, and Roland Vollgraf. "Fashion-MNIST: a Novel Image Dataset for Benchmarking Machine Learning Algorithms". In: (2017). DOI: 10.48550/ARXIV.1708.07747. URL: https://arxiv.org/abs/1708.07747 (cit. on p. 5).