Defect detection MultiHeadAttention Fusion model on images acquired with different light sources

Michele Somero
DMIF
University of Udine
Udine, Italy
0000-0001-5846-8827

Federico Urli
DMIF
University of Udine
Udine, Italy
0000-0001-9861-9677

Lauro Snidaro
DMIF
University of Udine
Udine, Italy
0000-0003-3828-9017

Alessandro Liani
Video Systems Srl
Viale Mangiarotti n.4, 33033
Codroipo (UD), Italy
0000-0001-9570-7727

***Abstract*—In this paper, we discuss and try a multi-image fusion approach with a multibranch Convolutional Neural Network (CNN) that implies a MultiHeadAttention (MHA) technique in the fusion center. This work studies the employment of the architecture on actual data containing different images of the same USB device. The images differ in the direction of the light at the moment of the acquisition. We observed that instead of employing a simple concatenation fusion of the outputs, the network architecture could employ a multibranch classification featurewise, which utilizes a multi-head attention mechanism instead of a channel attention one.**

***Index Terms*—Fusion, Defect Detection, MultiHeadAttention, Attention, Illumination, Multiple Inputs**

I. Introduction

Defect detection in industry is a big challenge in the Deep Learning field. Specifically, in mechanical production or steel manufacturing, the amount of defective material is substantial, and it can affect the whole production chain if the defects are not removed or corrected at the beginning of the process [1]. Since the defect detection problem is well studied, multiple attempts in this field have been made, mainly employing Machine Learning and Deep Learning [1], [2]. The possible solutions utilize Convolutional Neural Networks (CNN), work on images of the possible defective piece, and try to extract the defect or predict its condition. One example of a defect-detection study is the well-known Steel dataset Severstal [3], vastly studied with different techniques of Deep Learning like the mixed supervision of two sub-networks by [4].

In some cases, the acquisition possibilities increase, and several images of the same objects can be analyzed simultaneously. In these circumstances, the possible solutions include: the fusion of the images at the beginning of the Deep Learning process or a fusion of the extracted features in the process itself in order to obtain a classification [5].

Our work focuses on a specific production problem previously studied by [6]: USB stick defects detection. This problem occurs on the highly reflective stainless steel USB connector, where several damages can happen, such as scratches or dents. The quality control behind these processes can highly impact the USB sticks' production and their functionality.

The idea behind the work of [6] relies on their dataset, composed of multiple groups of images of USB stick connectors. Analogous approaches have been made in similar fields: in [7], the core idea is to combine bright-field and dark-field illumination images of the same object and feed them to a patch aware feature matching network so that the usage of two different illumination angle reveals more contrasted images in the defected regions due to the roughness of the material. Moreover, [8] focuses on the segmentation of some defective gears, washers, and screws. The authors of this paper developed an illumination chamber for image acquisition. They collected several images of the same object with highly directional and contrasted images to fuse the images and detect the defects' location upon the parts. Lastly, [9] developed a multi-lights source lightning strategy employing reinforcement learning to meliorate the detection of little particular defects that could not be detected with a single inspection.

Our work focuses on a multi-illumination image fusion technique as in the cited studies. This paper proposes a network architecture that relies on four base models fine-tuned to detect each different image and combined with a multilayer-attention technique, which serves as a feature correlation between the models before the actual decision process. The model was trained on a multi-label classification task with a dataset published by [6].

The paper is organized as follows: Section II reviews the dataset and the preprocessing techniques applied to the images; Section III presents the overall architecture of the network focusing on the core elements, which are the four networks fine-tuned on the different images. Section IV describes the fusion center with the MultiHeadAttention layer employed for combining the networks. Section V and VI describe respectively the experiments and the results. Conclusions and indications for future work are given in Section VII.

II. Dataset Explanation

The dataset employed in all the experiments comes directly from the authors of [6], who shared part of the dataset discussed in their paper on a GitHub page[1].

The data we could download comprehend ten different images of 687 different USB sticks, each employing a different light source positioned adjacent to the pen drive connector

[1]https://github.com/Xavierman/Fusion-of-multi-light-source-illuminated-images-for-defect-inspection/tree/main

Dataset classification defects and numbers						
Defect type	**Bright Line**	**Deformation**	**Dent**	**Normal**	**Scratch**	**Total**
Number of images	112	116	180	51	228	687

TABLE I: Number of samples per defect present in the dataset created by [6]. Each sample contains ten images of the same USB stick where the light position differs. The total number of images is 6870. As expressed in sec. II, only the ”high light” images were used.

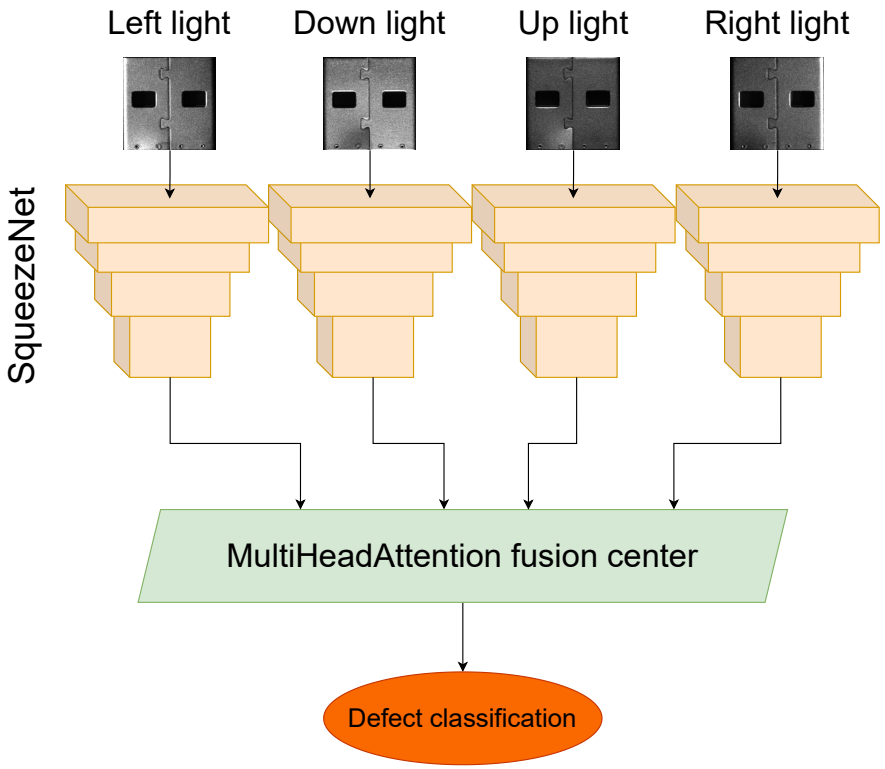
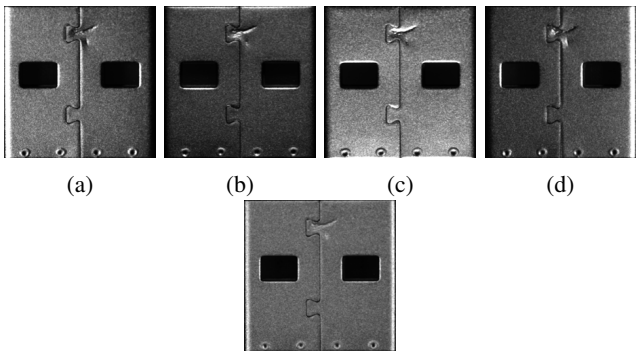


Fig. 1: Architecture of the overall fusion model, which employs four SqueezeNets as base networks for each branch.

SqueezeNet Fine Tuning Results (3-fold)					
Light Direction / Metrics	**Down**	**Left**	**Right**	**Up**	**Uniform**
Accuracy	0.809 (±0.06)	0.888 (±0.02)	0.838 (±0.02)	0.818 (±0.02)	0.790 (±0.03)

TABLE II: Accuracy results of the 3-fold cross-validation with a standard deviation of the finetuned SqueezeNet model on the four ”high light” images and the uniform one.

itself in one of the four cardinal directions with respect to the acquired object, as shown in Fig. 2. Unlike what [6] uses for their tests, the dataset did not include more than 700 samples (USB connectors), and the ones classified ”Normal” were not additional 900 samples but were included in the 687. The images have been labeled, as shown in table I, in five classes: BrightLine, Deformation, Dent, Scratch, and Normal. The first four categories comprehend all typical surface imperfection that damages the stick, but the last one represents the normal condition of the USB connector without defects. The whole dataset comprehended two main types of images: ”high light” and ”low light”. Each category comprehends the four illumination images but differs in brightness. For the experiments in this work, we decided to train the network on ”high light” images to align with the experiments by [6]. It is worth noting that the dataset includes an additional image per USB connector, where each stick has been acquired with a global, uniform illumination. While we trained and tested during our



(a) (b) (c) (d) (e)

Fig. 2: Example of five different images of the same defective USB connector acquired by [6] in different illumination conditions. In each image the position of the light is set in a different cardinal position with respect with the object: (a) left, (b) up, (c) down, (d) right. The last USB connector (e) was acquired with a uniform light from above the object.

experiments on this image, we followed the methodology of [6] and did not include the uniform image in the fusion techniques.

The image preprocessing for the fine-tuning was a relatively simple process. Each image was first scaled from a range [0,255] to a range [0,1]. Then, the dataset was adjusted to have its mean equal to 0 feature-wise. Finally, the dataset was divided into training and test sets. During the training of the fusion models, we performed a slight data augmentation to reduce the network overfitting. This function comprehended a random flip function, a random brightness function with a 0.3 deviation, and a random contrast function with [0.75, 1.25] as the contrast range.

III. Architecture of the Model

The architecture the paper expresses is based on the idea from [6], where the authors employed a single architecture, fine tuned on the images, and combined the four results on each light direction in a final one, using a channel attention fusion, that resulted in improving the overall accuracy. This architecture, instead, takes the core of their architecture, the Squeezenet, and attempts to fuse the results in a multi-head way. Each network outputs a tensor, then compared to the others with a Multi-Head Attention layer: the tensor of one branch is compared with a combination of the other 3. This comparison should allow the network to take the most valuable elements from each branch and combine them with the others to increase the model's accuracy.

The overall architecture, as shown in Fig. 1, comprises four base models, which are singularly trained in a specific light direction. Each base model, represented as a SqueezeNet, outputs a classification, and, at an earlier stage, a tensor extracts the information from that network and fuses it in the fusion center. This last block is in charge of fusing all the information, caring for the similarities between the images, and outputting a final combined classification.

A. Squeezenet

The advancement of multiple CNN classifiers in the past years led to multiple choices for high-accuracy image classification networks, but instead of focusing on GoogLeNet [10] or ResNet [11], we, in line with the work of [6], decided to work with the lighter SqueezeNet [12]. Furthermore, our architecture needed to operate four equal models in parallel, and this smaller but powerful network could be utilized without sacrificing the quality, the training time, or the training resources.

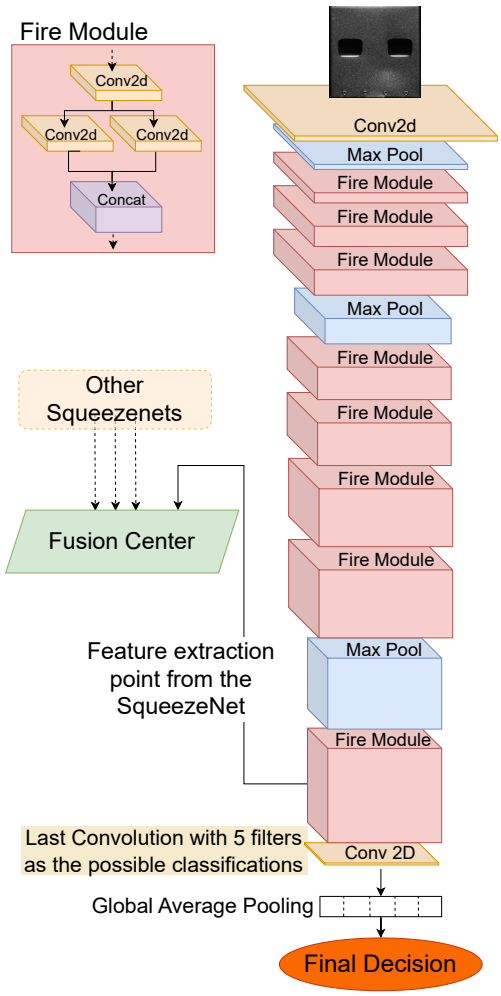


Fig. 3: Architecture of the SqueeezeNet by [12] architecture employed as base network for each branch of the fusion model.

The SqueezeNet employed in this paper comes from the original SqueezeNet by [12]. The base architecture is similar to the original one, with some tweaking in between, similar to what [6] did in their paper. The model employs a L2 regularizer of 0.0001 on the Convolutional layers and a final 10% Droput layer to diminish some training overfitting. The models, pre-trained on ImageNet, have been converted into tensorflow and finetuned on the dataset. This training employed a softmax final activation and a Categorical Cross-

Fusion Details			
Layers	**Input layer**	**Output size**	**Implementation Notes**
MHA for input 1	[Output SqueezNet 1, Concatenation of the other 3 outputs]	(12,12,512)	heads: 3, key dim: 64, value dim: 64
MHA for input 2	[Output SqueezNet 2, Concatenation of the other 3 outputs]	(12,12,512)	heads: 3, key dim: 64, value dim: 64
MHA for input 3	[Output SqueezNet 3, Concatenation of the other 3 outputs]	(12,12,512)	heads: 3, key dim: 64, value dim: 64
MHA for input 4	[Output SqueezNet 4, Concatenation of the other 3 outputs]	(12,12,512)	heads: 3, key dim: 64, value dim: 64
LayerNormalization 1	MHA for input 1	(12,12,512)	Dropout: 30%
LayerNormalization 2	MHA for input 2	(12,12,512)	Dropout: 30%
LayerNormalization 3	MHA for input 3	(12,12,512)	Dropout: 30%
LayerNormalization 4	MHA for input 4	(12,12,512)	Dropout: 30%
Concatenation layer	[LayerNormalization 1, LayerNormalization 2, LayerNormalization 3, LayerNormalization 4]	(12,12,2048)	axis = -1
Channel Attention Layer	Concatenation layer	(10,10,2048)	Dropout: 10%
Convolutional 2D layer 1	Channel Attention Layer	(10,10,512)	filter size: 512x3x3, stride: 1, activation: RelU
Convolutional 2D layer 2	Convolutional 2D layer 1	(10,10,5)	filter size: 5x1x1, stride: 1, activation: RelU
Global Average Pooling	Convolutional 2D layer 2	(1,5)	

TABLE III: Detailed configuration of the final stage of the fusion architecture as in Fig. 4.

entropy loss. The results of the finetuning training are shown in table II.

Along with the four SqueezeNet trained as base branches for the fusion network, another instance of this model has been finetuned on the uniform images to compare the result with the other SqueezeNets and the final results.

IV. MultiHeadAttention fusion center

There are several methods of fusing data and models in a Neural Network architecture. The most common include the average fusion of the final results, the combination of the features with a final decision network, and decision fusion. In [6] case, the fusion is composed of a simple classifier with a concatenation that employs a ChannelAttention method on each branch to perform a re-calibration of each tensor before the concatenation.

In our case, we decided to follow a transformer-based fusion where each branch is correlated with the others thanks to a MultiHeadAttention (MHA) layer, as expressed in [13]. The MultiHeadAttention technique allows the model to collect information from different branches. As in [13], the concept is to attend to the information from different positions and fuse them.

$$MHA(Q,K,V) = Concat(head_1,...,head_h)W^O$$
$$\text{where } head_i = Attention(QV_i^Q, KW_i^K, VW_i^V) \quad (1)$$

Equation 1 by [13], represents the MultiHeadAttention layer employed in this paper where the Query (Q) is compared

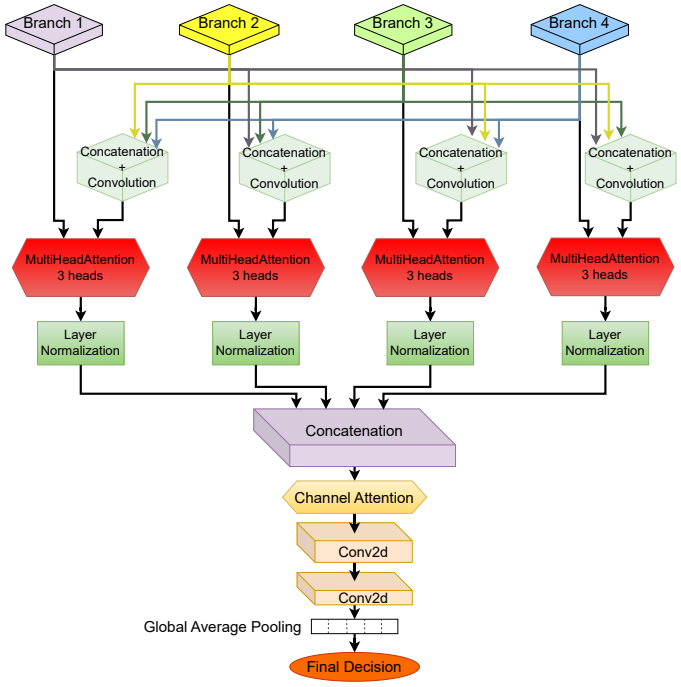


Fig. 4: MultiHeadAttention fusion center. Each output branch is combined with the others according to a MultiHeadAttention layer.

Model fusion evaluation and comparison (3-fold)					
Model Name / Metrics	**Uniform illumination**	**Average decision fusion**	**Combination w decision module**	**CA re-calibration***	**MHA Fusion (Ours)**
Accuracy	0.790 (±0.03)	0.872 (±0.02)	0.878 (±0.02)	0.886 (±0.03)	**0.901 (±0.04)**
F1	0.790 (±0.03)	0.870 (±0.02)	0.867 (±0.02)	0.887 (±0.03)	**0.903 (±0.04)**

TABLE IV: Accuracy and F1 score of the 3-fold cross-validation with a standard deviation of the fusion models. The comparison comprehends the Uniform SqueezeNet, trained on the "uniform light" images.
*Note that the Channel Attention (CA) re-calibration model is based on [6] model. It has the same architecture, but the data in this table results from training on our actual data (different from the one described in their study).

with the Value (V) according to the Key (K). This process produces an attention probability matrix based on some trained Weights (W), which should highlight similar features between Query and Value. Specifically, in our network architecture, as illustrated in Fig. 4, each output of the four branches is fed to a MultiHeadAttention layer along with a combination of the other three to determine the best similarities between the information extracted by each branch. The core idea is to calibrate the concatenation inputs concerning the other three branches, correlating the similar features and the different ones. The MHA layer is then normalized with a LayerNormalization; moreover, a Dropout layer is applied before the concatenation of each tensor to avoid overfitting behaviors. The MultiHeadAttention layers are trained with three attention heads, so the three parallel attention layers have the same value of 64 for the size of both the attention of the query and values.

The result of the concatenation is computed as the authors of [6] did: a final ChannelAttention layer to calibrate the output, a Convolution to scale to a fourth of the features and a final Convolution and a GlobalAveragePooling are computed to output the final classification of the five possible classes. Table III describes in detail the implementation of this fusion center.

V. Experiments

All the experiments described in this paper utilize the 3-fold cross-validation technique. We developed several investigations during our tests in order to compare our results with something as similar as possible to the state of the art. We developed base experimentations: an average decision fusion and a feature branch combination with a decision module. Furthermore, we trained an architecture as similar as possible to the one of [6] to train on our specific data.

As clarified in sec. II, the experiments employed half of the USB multi-light source illuminated dataset that can be found on the GitHub page[2] of the paper [6].

The training process for each fusion model took at most 500 epochs with an early stopping routine after 75 epochs without an improvement in the loss value. For the SqueezeNets, the training process was set to a maximum of 4000 epochs and an early stopping of 200.

[2]https://github.com/Xavierman/Fusion-of-multi-light-source-illuminated-images-for-defect-inspection/tree/main

A. Branches finetuning

As explained in III-A, the SqueezeNet base models, pretrained on the ImageNet dataset [14], have been finetuned on the multi-light source illuminated dataset singularly. Each network has been trained on a single light direction to focus the classification problem under that specific light condition. The idea was to transform each model into an expert in that specific light condition. Then, we set the weights on that state so that, when combined, the outcome could be a combination of knowledge of each model. The results of the finetuning can be seen in table II.

B. Fusion center training

All the fusion techniques involve using the four finetuned models without further training. The weights are frozen, and the fusion happens in the late stage of the models: according to [6], the best extraction point for the features from the SqueezeNet is the second-last convolution. The training employs an ImageDataGenerator with a data augmentation function, which was explained previously in the paper. Furthermore, the model utilizes a Stochastic Gradient Descent Optimizer (SGD) algorithm, which employs a learning rate and momentum of 0.0025 and 0.9. The optimization has been made utilizing a categorical cross-entropy loss.

VI. Results

In this study, we decided to implement a different fusion approach from the one of [6]. The MultiHeadAttention layer lets the model compare the tensors from each branch and weights the branches during the fusion. This approach led to the training results summarized in table IV. The overall accuracy of the fusion proved to be better than that of the other approaches we tried. Moreover, the implementation of the model from [6] we tested on the data we had, proved slightly less accurate than our implementation. An additional test compared the fusion model with a single model trained on a single image captured in a uniformly lighted environment (Uniform illumination). However, the results of this approach were both less effective than all fusions and single models, as can be seen in table IV and table II; in fact, the MHA Fusion model gained almost 14% of the accuracy. In general, our approach proved to be 8% more accurate on average than relying on a single model. Notably, the single model trained

on the Left light images has high accuracy in detecting many defects in the images and proved to be better than some of the fused models. This behavior could impact the actual values of the final combinations, but it is unlikely that it could impact the results of the comparison between the fusion models.

ACKNOWLEDGEMENTS

This publication was realized with the co-funding of the European Union - ESF REACT-EU, PON Research and Innovation 2014-2020.

VII. CONCLUSIONS

This paper studies the feasibility of a MultiHeadAttention fusion of multiple images that differ in light conditions. The study shows the possibility of conditioning the fusion thanks to the search for correlation between the acquisitions. Our results exhibit an impact of this fusion technique on the model's accuracy, especially considering the employment of a single uniform illumination of the analyzed object.

The possibility of utilizing this fusion technique impacts future works on multiple light condition image combinations. Furthermore, our future work will focus on developing reinforcement learning architectures for defect detection, like in [9], that use the proposed fusion method to enhance the accuracy in classifying and segmenting defective objects acquired with different illumination conditions.

REFERENCES

[1] A. A. P. Chazhoor, E. S. L. Ho, B. Gao, and W. L. Woo, "A Review and Benchmark on State-of-the-Art Steel Defects Detection," *SN Computer Science*, vol. 5, p. 114, Dec. 2023.

[2] P. M. Bhatt, R. K. Malhan, P. Rajendran, B. C. Shah, S. Thakar, Y. J. Yoon, and S. K. Gupta, "Image-Based Surface Defect Detection Using Deep Learning: A Review," *Journal of Computing and Information Science in Engineering*, vol. 21, p. 040801, 02 2021.

[3] A. Grishin, BorisV, iBardintsev, inversion, and Oleg, "Severstal: Steel defect detection," 2019.

[4] J. Božič, D. Tabernik, and D. Skočaj, "Mixed supervision for surface-defect detection: From weakly to fully supervised learning," *Computers in Industry*, vol. 129, p. 103459, 2021.

[5] Y. Xu, Z. Li, Z. Jiang, T. Wang, H. Wang, Y. Zhai, and Q. Cen, "Defect recognition method based on fusion learning of multi-layer image features," in *2022 IEEE 4th International Conference on Power, Intelligent Computing and Systems (ICPICS)*, pp. 342–345, 2022.

[6] G. Fu, S. Jia, W. Zhu, J. Yang, Y. Cao, M. Y. Yang, and Y. Cao, "Fusion of multi-light source illuminated images for effective defect inspection on highly reflective surfaces," *Mechanical Systems and Signal Processing*, vol. 175, p. 109109, Aug. 2022.

[7] Q. Sun, K. Xu, H. Liu, and J. Wang, "Unsupervised surface defect detection of aluminum sheets with combined bright-field and dark-field illumination," *Optics and Lasers in Engineering*, vol. 168, p. 107674, 2023.

[8] D. Honzátko, E. Türetken, S. A. Bigdeli, L. A. Dunbar, and P. Fua, "Defect segmentation for multi-illumination quality control systems," *Machine Vision and Applications*, vol. 32, p. 118, Nov. 2021.

[9] C.-K. Cheng and H.-Y. Tsai, "Enhanced detection of diverse defects by developing lighting strategies using multiple light sources based on reinforcement learning," *Journal of Intelligent Manufacturing*, vol. 33, pp. 2357–2369, Dec. 2022.

[10] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, "Going deeper with convolutions," in *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1–9, 2015.

[11] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 770–778, 2016.

[12] F. N. Iandola, S. Han, M. W. Moskewicz, K. Ashraf, W. J. Dally, and K. Keutzer, "SqueezeNet: AlexNet-level accuracy with 50x fewer parameters and <0.5MB model size," Nov. 2016. arXiv:1602.07360 [cs].

[13] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention is all you need," 2023.

[14] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "Imagenet: A large-scale hierarchical image database," in *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 248–255, 2009.