ENHANCING THE SAFETY OBJECT DETECTION ACCURACY IN CONSTRUCTION SITE USING A FREQUENCY CHANNEL ATTENTION NETWORK LAYER

A Thesis

by

INUK KANG

Submitted to the Graduate and Professional School of
Texas A&M University
in partial fulfillment of the requirements for the degree of

MASTER OF SCIENCE

Chair of Committee,	Zixiang Xiong
Committee Members,	Nima Kalantari
	Serap Savari
	Ulisses Braga Neto
Head of Department,	Miroslav Begovic

August 2022

Major Subject: Electrical Engineering

ABSTRACT

Currently, research on securing safety by unmanned systems is being actively conducted. Development is underway to reduce costs and secure worker safety by filling safety-related personnel's blind spots and reducing their burden. For intelligent safety security, we propose artificial intelligence models that can detect, identify and distinguish major objects based on photographic information. In addition, Frequency Channel Attention Network (FcaNet), which supplements the existing Global Average Pooling (GAP) method, is used to improve the existing algorithm, and the accuracy is improved.

For this purpose, 12,000 pieces of photographic data images are collected for 5 major equipment to be encountered in the actual construction environment. The detection and identification performance of the model is maximized by using the FcaNet layer for learning through the existing Faster-RCNN, Libra-RCNN, and Double-Heads model. As a result, the accuracy of the test dataset is improved by 6%, 0.4%, and 0.4%, respectively. And, through using random initialization and improved batch normalization, the shortcomings of limited data are reduced, and the effect of pre-training is obtained without. This results in an improvement of more than 20% in each model, and the revised model shows 0.5% higher than the existing one. It is hoped that these results will be reflected in the work environment intelligence project to further reduce the burden on manpower and improve efficiency.

DEDICATION

To my Family, fahter Youngsoon Kang, mother Kyeongja Yoon and sister Hyeonseo Kang, for teaching me the most dedication.

To Republic of Korea Army for giving me the chance to study abroad and widen horizons.

CONTRIBUTORS AND FUNDING SOURCES

Contributors

This work was supported by a thesis committee consisting of Professor Zixiang Xiong of the Department of Electrical and Computer Engineering.

Funding Sources

There are no outside funding contributions to acknowledge related to the research and compilation of this thesis. The student gets funding for studying and staying in the United States from Republic of Korea Army (ROKA).

NOMENCLATURE

AP	Average Precision
CNN	Convolutional Neural Network
CSP	Center-and-Scale Prediction
DCT	Discrete Cosine Transform
FcaNet	Frequency Channel Attention Network
FPN	Feature Pyramid Network
GAP	Global Average Pooling
GN	Group Normalization
IoU	Intersection of Union
MR	Miss Rate
PPE	Personal Protection Equipment
RCNN	Region-based Convolutional Neural Network
RPN	Region Proposal Network
SE	Squeeze-and-Excitation
SyncBN	Synchronized Batch Noramalization

TABLE OF CONTENTS

LIST OF FIGURES

FIGURE Page

LIST OF TABLES

TABLE Page

1. INTRODUCTION

As an important material production sector and pillar of industry of the national economy, the construction industry plays an important role in improving living conditions, improving infrastructure, absorbing labor employment, and promoting economic growth. At the same time, it is also a high-risk industry with frequent safety accidents. Construction workers are exposed to a wide range of hazards, including physical (noise, extreme temperatures, slick floors), chemical (solvents, cement, respirable crystalline silica), mechanical (slips, falls, heavy tools, crushing), and ergonomic (repetitive tasks, awkward postures, overexertion, using the wrong tools) hazards, all of which put them at risk for a variety of occupational diseases. Personal Protection Equipment (PPE) is defined by the Occupational Safety and Health Administration (OSHA) as equipment used to reduce exposure to a variety of risks, and construction workers are advised to wear a variety of protective gears. Eye and facial protection (safety glasses, goggles, or a face shield), foot protection (safety shoes), hand protection (gloves), and head protection are all included [2]. According to a survey done by the United States Bureau of Labor Statistics (BLS), 84% of employees who had gotten head impact injuries were not using head protection equipment [3]. Sehsah et al. show that only 59.4% of workers wear PPE while at work. The most common reported reasons by non-users are uncomfortable (78.2%), lack of knowledge on how to use (73%), poor fit/falling off (69.2%), feeling too hot or unavailability (69.2%) [2]. This is simple but essential to safety management. The biggest problem is finding it, and that is where the most manpower goes. In the safety management organization, supervisors are located at the site and perform the role of inspecting and controlling the wearing of PPE and actions of workers. However, this is laborious, and there are problems such as the existence of blind spots due to the lack of staff, the cost of the contractor for supervisors, and personal conflicts between administrators and workers. Therefore, there is an urgent need for a consistent and effective application of unmanned detection technology in the traditional way that relies on the human eyes.

For some years, the burgeoning area of Artificial Intelligence (AI) has been challenging the construction business. The deep usage of AI is rapidly moving the industry ahead to automation and autonomous systems, from planning to execution. Machine-generated project planning for estimating project costs or creating the site layout, automated monitoring of construction operations and worker safety and deployment of robots to execute construction jobs are just a few examples [4]. AI is attempting to approximate traditionally difficult issues using human-inspired algorithms. The following are the most significant benefits of AI for business: using the data it offers, improving end-user experience, automating processes to allow workers to focus on work that adds value, decreasing human mistakes and providing services more rapidly. This may be observed in connection to lean concepts, which emphasizing maximizing value, eliminating waste, and improving working process efficiency [5]. Furthermore, recording equipment and technology that are linked to a variety of networks [6] are moving toward more efficient, resource-saving, but securing a lot of information in manpower-intensive construction environment.

Although object detection in the traditional computer vision field has a very mature technology for detecting some specific targets, the algorithms often do not perform well when directly applied to the construction site to detect workers and PPE. First, the targets captured by the imaging sensor are too small to locate because the surveillance cameras are often positioned at a relatively high position to comprehensively monitor the operation process of the construction site. Second, the sizes of the PPE are smaller than the ones of the operator and are easy to confuse with the operator itself, which further increases the difficulty of detection. Therefore, how to improve the performance of detecting PPE on construction sites has become a challenge. There are several attempts to overcome this problem. Liu et al. conduct a comprehensive evaluation of the field linked to the use of computer vision technologies to monitor construction workers' dangerous conduct. They use classic machine learning and deep learning methods to investigate the use of object detecting technologies in greater depth [6]. Fang et al. propose an object detection of Non-Hardhat-Use detection in far-field surveillance videos on construction sites. They study the various visual circumstances of building sites and classify image frames according to their visual conditions to test the method's applicability to the construction environment. They are then fed into the Faster-

RCNN model, which is divided into visual categories [7]. Nath et al. investigate YOLO-based CNN models for quick construction item identification. They use image dataset containing about 3,500 images and approximately 11,500 instances of common construction site objects. They evaluate object detection agility using the YOLO-v2 and YOLO-v3 models. Their method may also be expanded to accurately anticipate the relative distance of observed objects [4]. However, their methods are lacking in two ways. First of all, we find that the advanced object detection models are not used in detection. They use Faster-RCNN, a model that is the basis of object detection, is a fairly outdated model released in 2015. Although YOLO-v3 is recently developed, it focuses only on the detecting speed and exposes a serious flaw in accuracy. This makes it difficult to distinguish between wearing PPE or not and ultimately does not guarantee safety. Second, a large enough dataset is not used. This can lead to overfitting, resulting in high variance and very high error on a test set. Also, transfer learning or pre-trained models can help speed up convergence but do not consequently improve accuracy if the target dataset is too small.

We implement this research by improving three parts. First of all, in object detection model derived by increasing the accuracy in different ways, we think about a "magic key" that can improve all of them. As finding the detection necks generally use a Feature Pyramid Network (FPN) to extract semantic information, we modify the detection neck to make it have a stronger ability to extract semantic information. We can add the FcaNet layer [8] before each stage in neck outputs to the multi-scale features of the FPN in cutting-edge object detection models. The FcaNet layer that accepts input from various channels is selected to compensate for the disadvantage of not using the various inputs of the information of the existing GAP. Second, we collect 12,000 construction environment images with setting five classifications: Person, Head, Helmet, Jacket, and Red-life-jacket. We get different types of images: photos posted on the internet keyframes from surveillance cameras, real construction sites, and simulated wearing PPE photos. Also, to increase the difficulty of training and test, photos that are not related to the construction site or PPE are included. Third, although about 10k images are obtained, measures are needed to secure accuracy and prevent overfitting. To solve this, we train using scratch with normalization technique appropriately for optimization and training models for sufficiently long time to compensate for the lack of pre-training.

We get three results. First of all, the accuracy of the existing model with the FcaNet layer using Cityscapes dataset is improved by 0.5%. Second, we train and test three models with the FcaNet layer using our customized dataset, and the accuracies are improved by 6%, 0.5% and 0.4%, respectively. Third, using a customized dataset, scratch enhances accuracy by more than 20% compared to when it is not used. And, the revised model using the FcaNet layer gets an improvement of 0.5% additionally.

2. RELATED WORKS

2.1 Faster-RCNN

Faster-RCNN creates a Region Proposal Network (RPN) by adding a few more convolutional layers that regress region bounds and objectness scores at each place on a normal grid at the same time. A region proposal algorithm generates bounding boxes or locations of possible objects in the image, a feature generation stage obtains features of these objects, a classification layer predicts which class this object belongs to, and a regression layer refines the coordinates of the object bounding box. It includes a new anchor box that may be used as a reference at various scales and aspect ratios. It provides a pyramid of regression references in the regression layer, reducing the need to enumerate pictures or filters with numerous scales or aspect ratios. Figure 2.1 illustrates the RPN and RoI of Faster-RCNN.

Anchor's translation invariance is an important characteristic. First and foremost, anchors and functions do not require translation. Second, translation-invariance minimizes the size of the model, reducing the danger of overfitting on small datasets. It is also more cost-effective to use sliding windows with numerous sizes on feature maps. It creates a filter pyramid, classifies and regresses bounding boxes using anchor boxes of various sizes and aspect ratios. Users employ convolutional features calculated on a single-scale image as a result of this [9].

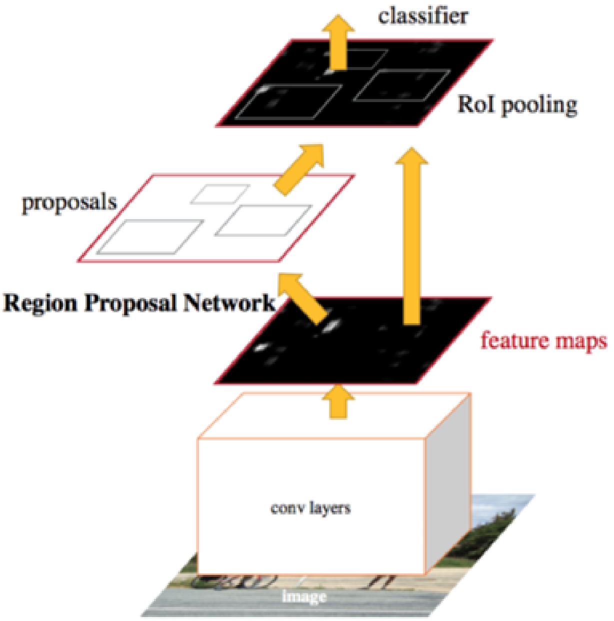

Figure 2.1: The Architecture of Faster-RCNN: Region Proposal Network and RoI pooling. Reprinted from S. Ren, K. He, R. Girshick, and J. Sun, “Faster r-cnn: Towards real-time object detection with region proposal networks,” Advances in neural information processing systems, vol. 28, 2015.

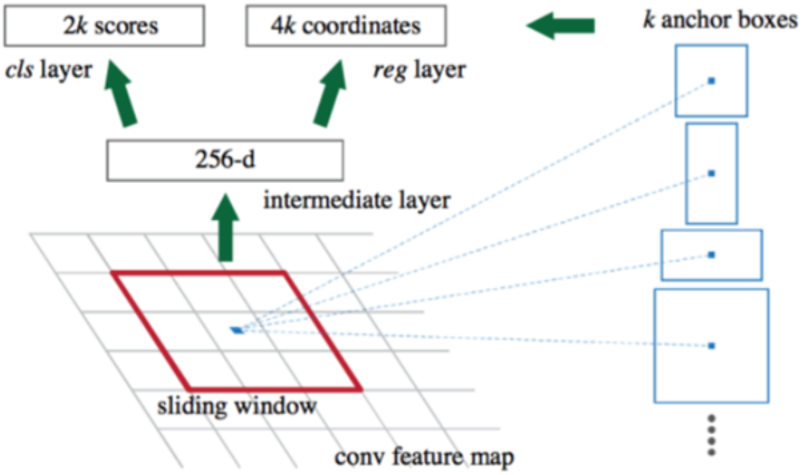

Figure 2.2: Region Proposal Network from different size of anchor boxes. Reprinted from S. Ren, K. He, R. Girshick, and J. Sun, “Faster r-cnn: Towards real-time object detection with region proposal networks,” Advances in neural information processing systems, vol. 28, 2015.

2.2 Libra-RCNN

For object detection, Libra-RCNN promotes balanced learning. IoU balancd sampling, balanced feature pyramid, and balanced L1 loss are all part of it. First, as an alternate technique for hard positive samples, it samples equal positive samples for each ground truth. The balanced feature pyramid is then used to create balanced semantic features by basic averaging features. These characteristics have been rescaled to make the original features stronger. Embedded Gaussian non-local attention is also used to enhance the balanced semantic characteristic. It simultaneously gathers low-level and high-level characteristics. Critical regression gradient is promoted with balanced L1 loss. This combines the gradient formulation and produces a balanced L1 loss. Figure 2.3 shows the architecture of Libra-RCNN.

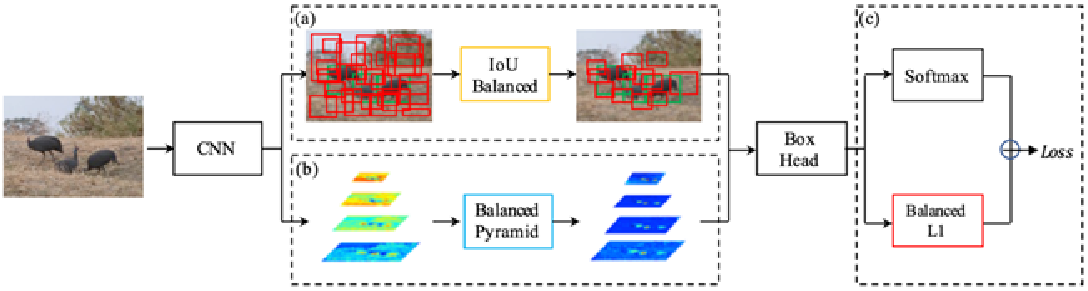

Figure 2.3: The overview of Libra-RCNN. Reprinted from J. Pang, K. Chen, J. Shi, H. Feng, W. Ouyang, and D. Lin, "Libra r-cnn: Towards balanced learning for object detection," in Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pp. 821–830, 2019.

Hard negative samples are the most common difficulty in sampling. More than 60% of hard negatives have an overlap more than 0.05 in the majority of samples, yet random sampling only offers 30% of training samples that are greater than the same threshold. Extreme samples are used to tackle this problem, converting difficult samples into hundreds of simple samples. According to IoU, sampling intervals are uniformly separated into K bins to increase the chosen chance of hard negatives. As a result, each bin receives the equal number of negative samples from the M matching candidates. Experiment parameters p_k can be used to set K:

$$p_k = \frac{N}{K} * \frac{1}{M_k}, k \in [0, K) \tag{2.1}$$

To achieve a balanced feature pyramid, it uses the same deeply integrated balanced semantic features to reinforce multi-level features.

in multi-level features as L, C_l is featured at resolution level l, lowest and highest index for l_{min} and l_{max}. The balanced semantic feature C is generated by averaging when features are rescaled. These are rescaled by using the same but opposite method to enhance the original characteristics [10]:

$$C = \frac{1}{L} \sum_{l=l_{min}}^{l_{max}} C_l \tag{2.2}$$

2.3 Double-Heads

The fully connected head ($fc-head$) is found to be more suited for classification, whereas the convolution head ($conv - head$) is shown to be more ideal for localization. As a result, these two heads have come together to work on both categorization and bounding box regression. Y. Wu et al. examine each individual based on predetermined suggestions and IoUs. Because $fc-head$ is more spatially sensitive while $conv - head$ employs a shared transformation, respectively. The use of each head to leverage the advantages of two head structures is demonstrated in the Double-Heads. In Figure 2.4, classification and localization are shared between a $fc - head$ and a $conv - head$, and Double-Heads is extended by incorporating supervision from an unfocused task during training and pooling classification results from both heads during inference.

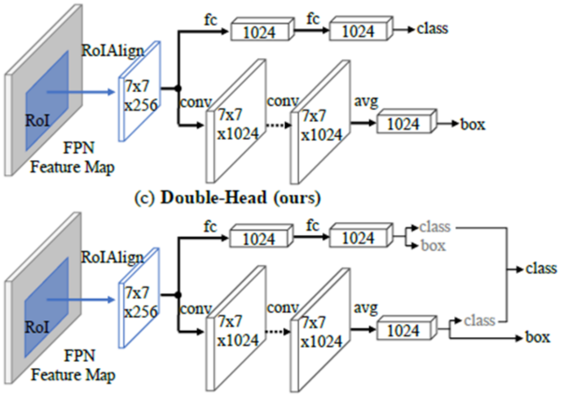

Figure 2.4: The overview of Double-Heads. reprinted from Y. Wu, Y. Chen, L. Yuan, Z. Liu, L. Wang, H. Li, and Y. Fu, "Rethinking classification and localization for object detection," in Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pp. 10186–10195, 2020.

Because of their different structures, the two heads gather complementing information for object classification. The two classifiers can be combined as follows:

$$s = s^{fc} + s^{conv}(1 - s^{fc}) = s^{conv} + s^{fc}(1 - s^{conv}) \tag{2.3}$$

where s^{fc} and s^{conv} are classification score from $fc - head$ and $conv - head$, respectively. The difference between the first and second scores is the product of the second score and the first score's inverse. This fusion is only useful under certain circumstances, $\lambda^{fc} \neq 0$ and $\lambda^{conv} \neq 1$ [11].

2.4 Attention mechanism

2.4.1 Squeeze-and-Excitation Networks

The Squeeze-and-Excitation (SE) block can be used to improve the quality of representations produced by a network by explicitly modeling the interdependencies in the channels of its convolutional features. It allows the network to learn to use global information to selectively emphasize informative characteristics while suppressing less helpful ones, allowing it to undertake feature recalibration. Figure 2.5 depicts the construction of the SE block.

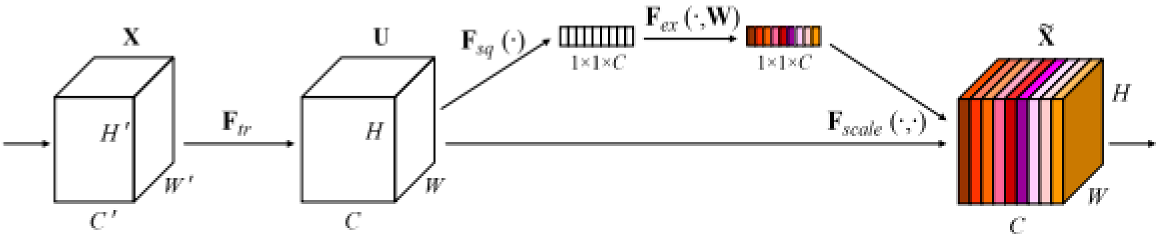

Figure 2.5: The overview of SENet. Reprinted from Hu, L. Shen, and G. Sun, “Squeeze-and-excitation networks,” in Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 7132–7141, 2018.

For any given transformation $\mathbf{F}_{tr}$ mapping the input $\mathbf{X}$ to the feature maps $\mathbf{U}$ where $\mathbf{U} \in R^{H\times W\times C}$, a convolution, it can construct a corresponding SE block to perform feature recalibration. The feature $\mathbf{U}$ is first squeezed, which results in a channel descriptor that aggregates feature maps across their spatial dimensions. An excitation operation, in the form of a simple self-gating mechanism, follows the aggregation. It takes the embedding as input and outputs a set of modulation weights for each channel. It can then be supplied straight into the network's succeeding tiers. The SE block is simple and may be immediately employed in existing state-of-the-art designs by replacing components with their SE counterparts, resulting in significant performance improvements. It is computationally light and adds just a little amount of model complexity and computational load.

Showing $\mathbf{F}_{tr}$ mapping an input $\mathbf{X} \in R^{H'\times W'\times C'}$ to feature maps $\mathbf{U} \in R^{H\times W\times C}$, the output as $\mathbf{U} = [\mathbf{u}_1, \mathbf{u}_2, \cdots, \mathbf{u}_C]$ can be written as:

$$\mathbf{u}_c = \mathbf{v}_c * \mathbf{X} = \sum_{s=1}^{C'} \mathbf{v}_c^s * \mathbf{x}^s \tag{2.4}$$

where $\mathbf{V} = [\mathbf{v}_1, \mathbf{v}_2, \cdots, \mathbf{v}_C]$ denotes the learned set of filter kernels. $\mathbf{v}_c$ is the parameter of the c-th filter. The output is generated by adding all channels together; channel dependencies are encoded in $\mathbf{v}_c$, but they are entangled with the local spatial correlation collected by the filter. It demonstrates that explicitly modeling channel interdependencies improves convolutional feature learning, allowing the network to boost its sensitivity to informative features. It gives the SE block access to global data and allows it to recalibrate the filter response.

It uses GAP to create channel-wise statistics z to aggregate information in the Squeeze stage. c-th element of $z \in R^C$ is calculated by

$$z_c = \mathbf{F}_{sq}(\mathbf{u}_c) = \frac{1}{H \times W} \sum_{i=1}^{H} \sum_{j=1}^{W} u_c(i, j) \tag{2.5}$$

It seeks to fully capture channel-wise dependencies in a second operation. To ensure that many channels can be emphasized, sigmoid activation is employed to meet flexibility and leaning non-mutually exclusive connection.

$$\mathbf{s} = \mathbf{F}_{ex} = \sigma(g(\mathbf{z}, \mathbf{W})) = \sigma(\mathbf{W}_2 \delta(\mathbf{W}_1, \mathbf{W}_1 \mathbf{z})) \tag{2.6}$$

wehre δ is the ReLu function, $\mathbf{W}_1 \in R^{\frac{C}{r} \times C}$ and $\mathbf{W}_2 \in R^{C \times \frac{C}{r}}$. ReLu and dimensionality-increasing layer return to the channel dimension of the transformation output $\mathbf{U}$. The block's final output is generated by rescaling $\mathbf{U}$ using the activations $\mathbf{s}$:

$$\widetilde{\mathbf{x}}_c = \mathbf{F}_{scale}(\mathbf{u}_c, s_c) = s_c \mathbf{u}_c \tag{2.7}$$

where $\widetilde{\mathbf{X}} = [\widetilde{\mathbf{x}}_1, \widetilde{\mathbf{x}}_2, \cdots, \widetilde{\mathbf{x}}_C]$ and $\mathbf{F}_{scale}(\mathbf{u}_c, s_c)$ refer to channel-wise multiplication between the scalar s_c and the feature map $\mathbf{u}_c \in R^{H \times W}$ [12].

2.4.2 Frequency Channel attention Layer

The GAP operation in the SE block is a pooling operation that is used to substitute completely linked layers in traditional CNNs. In the final convolutional layer, it creates one feature map for each matching category of the classification work. By imposing correspondences between feature maps and categories, it is more organic to the convolution structure. Furthermore, because the GAP has no parameters to optimize, overfitting is prevented at this layer. Furthermore, GAP sums up the geographical information, making it more resistant to input spatial translations.

The 2D DCT [13] can be written as:

$$G(k,w) = \frac{4}{MN}\sum_{m=0}^{M-1}\sum_{n=0}^{N-1} X(m)X(n)\text{cos}(\frac{k\pi}{M}(m+\frac{1}{2}))\text{cos}(\frac{w\pi}{N}(n+\frac{1}{2})) \tag{2.8}$$

suppose k and w are 0 in equation X, then it gets:

$$\begin{aligned} G(0,0) &= \frac{4}{MN}\sum_{m=0}^{M-1}\sum_{n=0}^{N-1} X(m)X(n)\text{cos}(\frac{0\cdot\pi}{M}(m+\frac{1}{2}))\text{cos}(\frac{0\cdot\pi}{N}(n+\frac{1}{2}) \\ &= \frac{4}{MN}\sum_{m=0}^{M-1}\sum_{n=0}^{N-1} X(m)X(n) \end{aligned} \tag{2.9}$$

Having a more in-depth mathematical analysis of this, we can see only $gap(X)HWB_{0,0}^{i,j}$ is utilized:

$$x_{i,j}^{2d} = gap(x^{2d})HWB_{0,0}^{i,j} + f_{0,1}^{2d}HWB_{0,0}^{i,j} + \cdots + f_{H-1,W-1}^{2d}HW{B_{0,0}}^{H-1,W-1} \tag{2.10}$$

$$s.t. i = 0, 1, \cdots, H-1, j = 0, 1, \cdots W-1$$

$$X = gap(X)HWB_{0,0}^{i,j} + f_{0,1}^{2d}HWB_{0,0}^{i,j} + \cdots + f_{H-1,W-1}^{2d}HWB_{0,0}^{H-1,W-1} \tag{2.11}$$

$$X_{i,:,:} = \begin{bmatrix} x_{0,0}^{2d} & \cdots & x_{0,W-1}^{2d} \\ \vdots & \ddots & \vdots \\ x_{H-1,0}^{2d} & \cdots & x_{H-1,W-1}^{2d} \end{bmatrix} = \begin{bmatrix} GB_{0,0}^{0,0} + D^{0,0} & \cdots & GB_{0,0}^{0,v} + D^{0,v} \\ \vdots & \ddots & \vdots \\ GB_{0,0}^{u,0} + D^{u,0} & \cdots & GB_{0,0}^{u,v} + D^{u,v} \end{bmatrix} \tag{2.12}$$

in which $X_{i,:,:}$ is the i-th channel of feature, $G = gap(X)HW, u = H-1, v = W-1$, and $D^{i,j} = f_{0,1}^{2d}B_{0,1}^{i,j} + \cdots + f_{H-1,W-1}^{2d}B_{H-1,W-1}^{i,j}$. This demonstrates that the conventional channel attention ignores all other frequency components and saves the lowest DC one. It has a proportionate relationship with GAP. It indicates that GAP corresponds to the lowest frequency, i.e., DC, component of 2D DCT, and GAP is a subset of 2D DCT. The channel attention mechanism uses just a tiny portion of the information in this way. As a result, this research extends GAP to include more AC coefficients in 2D DCT. Figures 2.6 and 2.7 show the existing GAP method and the revised one with the multi channel attention.

It divides input X into many parts along the channel dimension to employ various frequency components:

$$X^i \in R^{C' \times H \times W} i = 0, 1, \cdots, n-1. C' = \frac{C}{n} \tag{2.13}$$

And the 2D DCT results can be used as pre-processsing results of channel attention:

$$Freq^i = \text{2DDCT}^{u,v}(X^i) = \sum_{H-1}^{h=0} \sum_{W-1}^{w=0} X^i_{:,h,w} B^{u,v}_{h,w}, i = 0, 1, \cdots n-1 \tag{2.14}$$

The whole vector of pre-processing is achieved by concatenation of i. The multi-spectral channel attention may be represented as:

$$multispectral - attention = sigmoid(fc(Freq)) \tag{2.15}$$

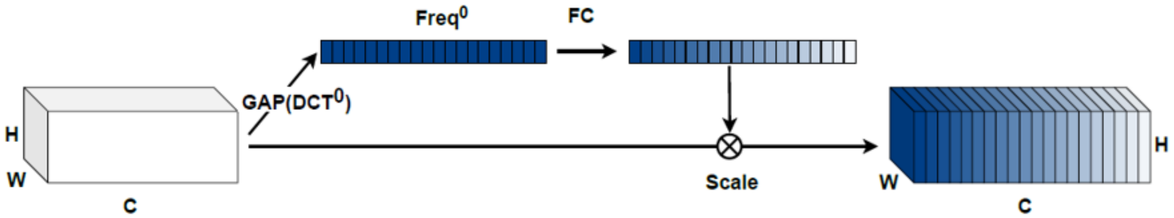

Figure 2.6: The overview of GAP frequency. Reprinted from Qin, Zequn, et al. "Fcanet: Frequency channel attention networks." Proceedings of the IEEE/CVF International Conference on Computer Vision. 2021.

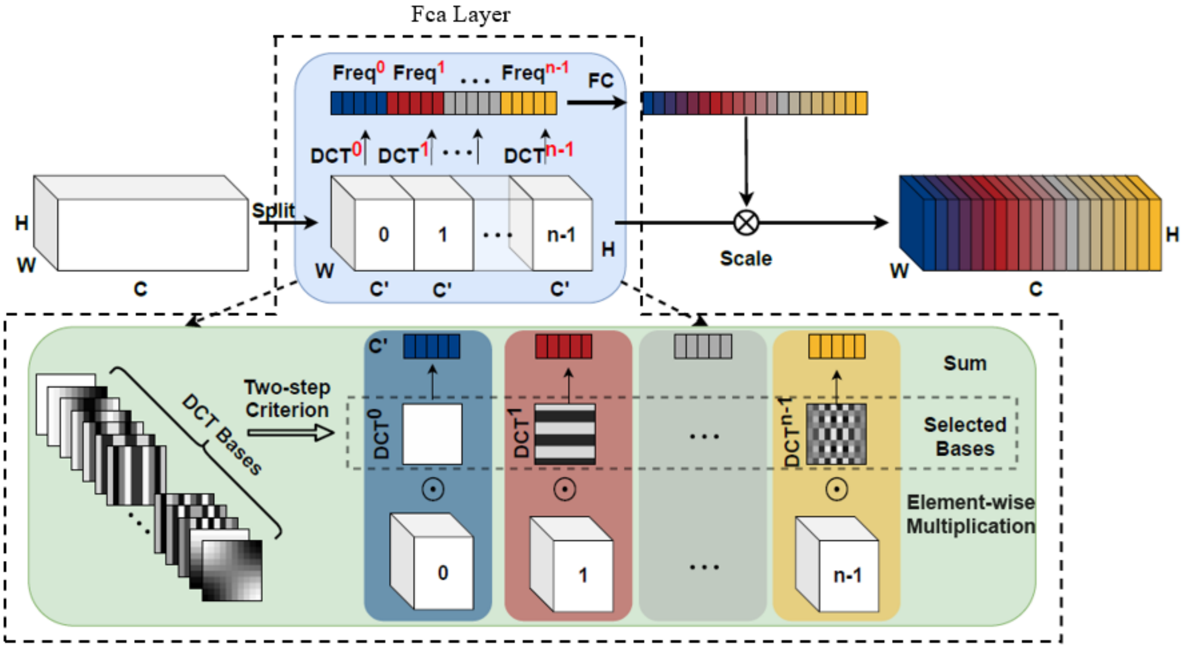

Figure 2.7: The overview of the multi channel attention frequency. modified from Qin, Zequn, et al. "Fcanet: Frequency channel attention networks." Proceedings of the IEEE/CVF International Conference on Computer Vision. 2021.

2.5 Using scratch dataset

The results of training on COCO from random initialization without any pre-training are close to those of pre-trained models. It is devoid of any pre-trained models or fine-tuning. This compensates for the lack of pretraining with revised normalization and extended training hours. The new normalization method employs Synchronized Batch Normalization (SyncBN) and Group Normalization (GN). GN provides computations that are unaffected by batch size or batch dimensions. SyncBN is used to implement Batch Normalization (BN). For BN, it avoids small batches and increases the effective batch size. In terms of training duration, there are no significant differences that random initialization yields less efficient results. Figure 2.8 shows the comparison of pre-trained model and random initialization [14].

GN separates channels into groups and normalizes the characteristics inside each group as a layer. It does not take use of the batch dimension, and its calculation is not affected by batch sizes

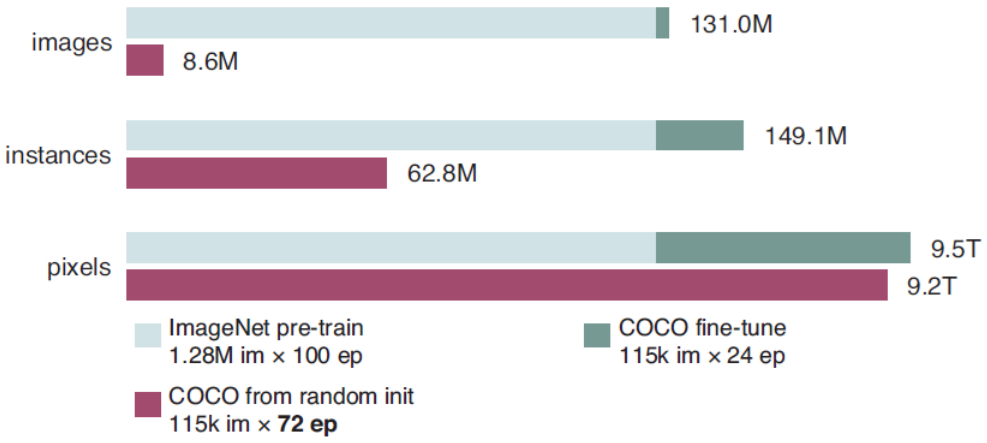

Figure 2.8: Comparing pre-training model and random initialization. Reprinted from He, R. Girshick, and P. Dollár, “Rethinking imagenet pre-training,” in Proceedings of the IEEE/CVF International Conference on Computer Vision, pp. 4918–4927, 2019.

[15]. The feature normalization method performs the following computation as in Equation 2.16:

$$\widehat{x}_i = \frac{1}{\sigma_i}(x_i - \mu_i) \tag{2.16}$$

μ and σ are the mean and standard deviation (std) computed by:

$$\mu_i = \frac{1}{m}\sum_{k\in S_i} x_k, \sigma_i = \sqrt{\frac{1}{m}\sum_{k\in S_i}(x_k - \mu_i)^2 + \epsilon} \tag{2.17}$$

with ϵ as a small constant, S_i is the set of pixels in which the mean and std are computed, m is the size of the set. In Batch Norm and Group Norm, the set S_i is defined respectively as:

$$S_i = \{k|k_C = i_C\}, \tag{2.18}$$

$$S_i = \{k|k_N = i_N, \left\lfloor\frac{k_C}{C/G}\right\rfloor = \left\lfloor\frac{i_C}{C/G}\right\rfloor\} \tag{2.19}$$

where G is the number of groups, which is a pre-defined hyper-parameter. C/G is the number of channels per group. $\lfloor\cdot\rfloor$ is the floor operation, and if each group of channels is stored in a sequential manner along the C axis, "$\left\lfloor \frac{k_C}{C/G} \right\rfloor = \left\lfloor \frac{i_C}{C/G} \right\rfloor$" signifies that the indexes i and k belong to the same group of channels.

Because the weights change dramatically in SyncBN, the linear scaling method may not be valid at the start of the training. As a result, it employs Linear Gradual Warmup [16]. It begins by setting the learning rate to a low value, such as r. The learning rate is then increased at a steady rate after each iteration until it reaches $\widehat{r}$. It can aid in good convergence when beginning training, but it is insufficient for bigger mini-batch sizes, such as 128 or 256. Then, Cross-GPU BN takes care of it. Given a total of n GPU devices, the sum value s_k is computed first using the training examples allocated to device k.

We get the mean value μ_B for the current mini-batch by averaging the sum values. Then, for each device, the variance σ_B^2 can be calculated. We can get standard normalization by $y = \gamma \frac{x-\mu_B}{\sqrt{\sigma_B^2+\epsilon}} + \beta$.

3. METHODOLOGY

A detector based on deep learning consists of three parts: backbone network, detection neck, and detection head. First, the backbone network extracts coarse features from the input image, then these features go through the detection neck to generate high-level semantic features, and finally, the detection head uses these high-level semantic features for classification and regression. Among them, to extract semantic information, the detecting neck often uses a FPN. As a result, we may make changes to the detecting neck to improve its capacity to extract semantic information. The FcaNet can be added to the multi-scale features of the feature pyramid network before the outputs.

3.1 Preliminary Works

Before applying different models and customized datasets, we preliminarily apply them to the existing dataset. Cityscapes is an image dataset with an emphasis on urban street scenes. This is usually used for autonomous driving, video surveillance, action recognition, and tracking. Since we judge that the research on the construction site environment to be carried out is more similar to that of MSCOCO, we decide to use it for the preliminary research. Also, we use a Center-and-Scale Prediction (CSP) model to take advantage of autonomous and intelligent surveillance. To make sure that the FcaNet layer has similar improvements with other data and models.

3.1.1 Center and Scale Prediction (CSP) Detector

In CNN-based approaches, broad object detection needs a sliding-window classifier or anchor-based predictions. It adheres to time-consuming window or anchor arrangements. It combines two issues: the location of the object and its size. However, the CSP detector proposes a higher-level abstraction that searches for object central points. It splits the "where" and "how" subproblems into two convolutions, each of which is expressed as a simple center and scale prediction. It produces a window-free, anchor-free environment, which makes training easier. As a result, it overcomes the constraints of anchor-based detectors and eliminates the time-consuming post-processing of key point pairing. Figure 3.1 shows the architecture of CSP model [17].

Detection in an anchor-based detector is usually stated as:

$$Dets = H(\Phi_{det}, B) = \{cls(\Phi_{det}, B), regr(\Phi_{det}, B)\} \tag{3.1}$$

where B is pre-defined based on the Φ_{det} collection of feature maps, and H is the detection head. $cls(.)$ and $regr(.)$ denote the prediction of classification scores and scaling, respectively, as well as the offsets of the anchor boxes. The anchor-free detector, on the other hand, can only be written using the detecting head and feature maps as:

$$Dets = H(\Phi_{det}) \tag{3.2}$$

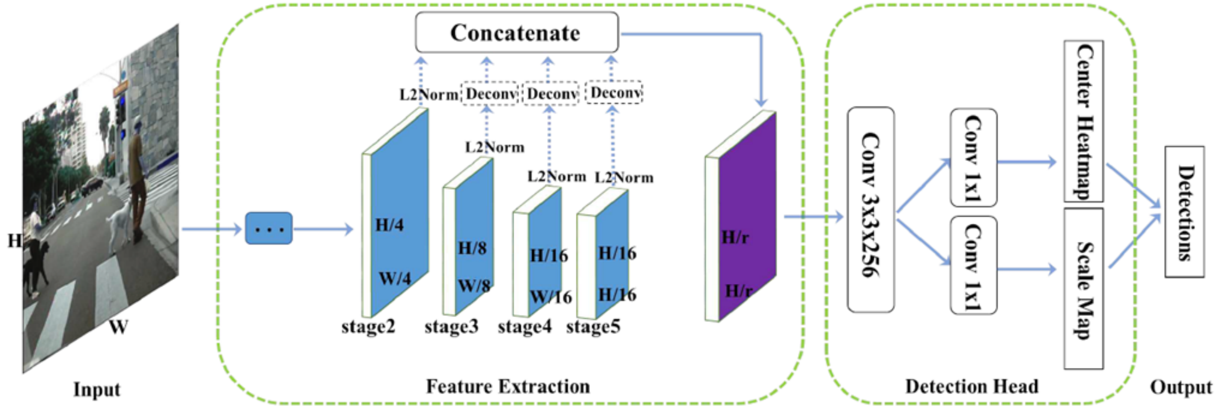

Figure 3.1: The overview of CSP Model. Reprinted from W. Liu, S. Liao, W. Ren, W. Hu, and Y. Yu, "High-level semantic feature detection: A new perspective for pedestrian detection," in Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 5187–5196, 2019.

To give more exact localization information, the feature extraction is separated into five steps via downsampling. Then, before concatenation and L2-normalization to rescale their norms to 10, the CSP model uses a deconvolution layer to create multi-scale feature maps with the same resolution. The two 1x1 $Conv$ layers in Detection Head provide the center heatmap and scale map, respectively.

When compared to merely employing the GAP in channel attention, using the multi-spectral attention has substantial performance gaps. The greatest result is obtained by using a configuration with 16 frequency components. In this thesis, we choose the frequency components that will give us the greatest results. Figure 3.2 depicts the visualization of frequency components. The FcaNet is added to the multi-scale features of the FPN before stage 3 to 5. Figure 3.3 depicts the new network structure.

We use the cross-entropy loss at the center prediction branch to frame the loss function as a classification task. A 2D Gaussian mask is centered at the site of each positive to eliminate the ambiguity of negatives surrounding the positives.

$$M_{ij} = \underset{k=1,2,\cdots,K}{max} G(i,j;x_k,y_k,\sigma_{w_k},\sigma_{h_k}), G(i,j;x_k,y_k,\sigma_{w_k},\sigma_{h_k}) = e^{-(\frac{(i-x)^2}{2\sigma_w^2}+\frac{(j-y)^2}{2\sigma_h^2})} \tag{3.3}$$

where K is the number of objects in a picture, (x_k, y_k, w_k, h_k) is the image's center coordinates, and the Gaussian mask's variance (σ_w^k, σ_h^k) is proportional to the height and width of individual objects.

The categorization loss may be represented as follows:

$$L_{center} = -\frac{1}{K}\sum_{i=1}^{W/r}\sum_{j=1}^{H/r}\alpha_{ij}(1-\hat{p}_{ij})^{\gamma}log(\hat{p}_{ij}) \tag{3.4}$$

where

$$\hat{p}_{ij} = \begin{cases} p_{ij} & \text{if } y_{ij} = 1 \\ 1-p_{ij} & \text{otherwise} \end{cases}, \alpha_{ij} = \begin{cases} 1 & \text{if } y_{ij} = 1 \\ (1-M_{ij})^{\beta} & \text{otherwise} \end{cases} \tag{3.5}$$

The network's estimated probability of whether or not there is an object's center in the position is $p_{ij} \in [0,1]$. $y_{ij} \in [0,1]$ is the ground truth label, one is the positive location. α_{ij} and γ are the focusing hyper-parameters.

For scale prediction, it is form of regression task with the smooth L1 loss:

$$L_{scale} = \frac{1}{K}\sum_{k=1}^{K} SmoothL1(s_k, t_k) \tag{3.6}$$

where s_k and t_k are the network's prediction and ground truth, respectively, for each positive. When the offset prediction branch is added, the smooth L1 loss is used in the same way.

the full optimization objective can be written as:

$$L = \lambda_c L_{center} + \lambda_s L_{scale} + \lambda_o L_{offset} \tag{3.7}$$

where λ_c, λ_s and λ_o are the weights for center classification, scale regression and offset regression loss.

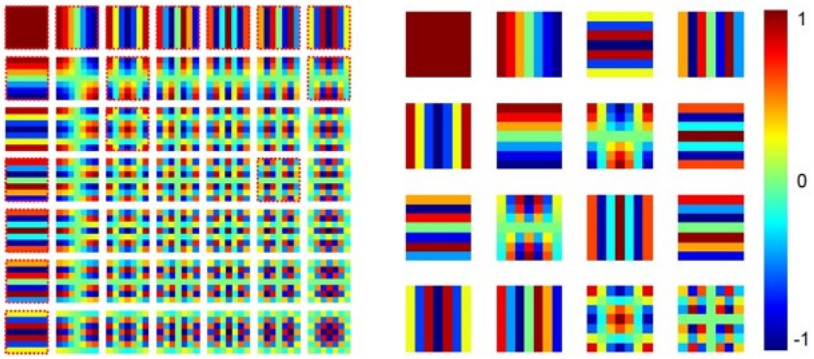

Figure 3.2: (Left) : The visualization of all frequency components. (Right) : Selected 16 frequency components. Reprinted from Qin, Zequn, et al. "Fcanet: Frequency channel attention networks." Proceedings of the IEEE/CVF International Conference on Computer Vision. 2021.

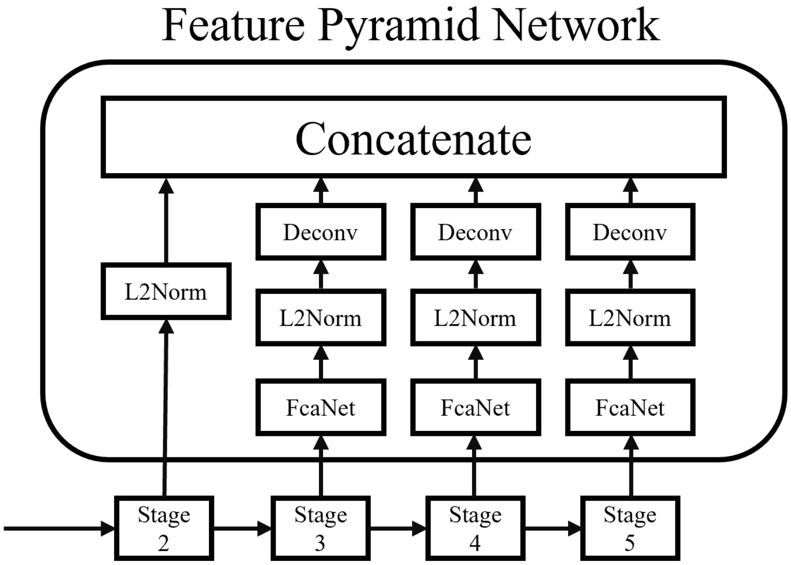

Figure 3.3: The revised CSP Model

3.1.2 Cityscapes dataset

The Cityscapes dataset is a large collection of data that focuses on the semantic understanding of the urban environment. It has been recorded in 50 different German cities. It contains around 31k annotated bounding boxes, and 2,975, 500, 1,575 photos in its training, validation, and testing sets. Figure 3.4 is an example of the Cityscapes image. It has photos at a number of places with the aim of decreasing city-specific overfitting, and collects photos over the course of many months, spanning spring, summer, and fall. The recordings are limited to excellent weather, which poses a substantial hurdle for computer vision and future extensions with customized datasets for varied weather circumstances [1]. We judge that the data were suitable for this thesis in image selection. MSCOCO dataset has the advantage of having a large number of images and annotations, but the disadvantage of a large number of images that are not suitable for the data environment has been highlighted. Table 3.1 shows a list of those that do not appear to be appropriate to training for object detection in this thesis.

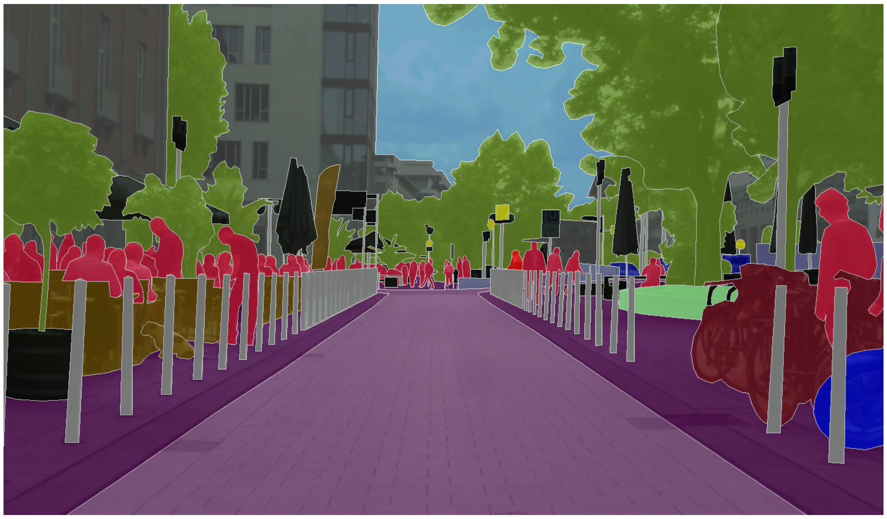

Figure 3.4: Example Image of Cityscapes

It chooses to concentrate on semantic, instance-wise dense pixel annotation since it is the most useful for training scene understanding algorithms, has the most extensive set of evaluations, and enables easy future dataset additions. It has 25 labels with various aspects, establishing a mix between common classes, varied uses, and "non-void" classes covering a vast portion of the image. The labels are described in Table 3.2.

Group	Classes
Sports Equipment	Frisbee, Snowboard, Surfboard, Skis, Balls, Bats, Skateboard, Tennis Racket, Kite, Baseball Glove
Food	Banana, Sandwich, Broccoli, Hot Dog, Donuts, Apple, Orange, Carrot, Pizza, Cake
Home Appliances	Chair, Sofa, Potted Plant, Dining Table, Toilet, Bed, Sink
Electronics	Laptop, Television, Computer Mouse, Microwave, Oven, Toaster, Remote Controller, Refrigerator, Cell Phone, Keyboard, Hair Drier
Objects	Bottle, Cup, Bowl, Wine Glass, Fork, Spoon, Book, Watch, Scissors, Pen

Table 3.1: Unnecessary Images in MSCOCO dataset

Group	Classes
Flat	Road, Sidewalk, Parking, Rail Track
Human	Person, Rider
Vehicle	Car, Truck, Bus, On Rails, Motorcycle, Bicycle, Caravan, Trailer
Construction	Building, Wall, Fence, Guard Rail, Bridge, Tunnel
Object	Pole, Pole Group, Traffic Sign, Traffic Light
Nature	Vegetation, Terrain
Sky	Sky

Table 3.2: Cityscapes Image annotations

3.1.3 Result of Preliminary Work

The detection performance is evaluated using log average Miss Rate (MR) over False Positive Per Image(FPPI) over the range $[10^{-2}, 10^{0}]$ denoted by MR^{-2}. It has different occlusion levels namely Reasonable, Small, Heavy, and All [18]. Table 3.3 shows the setting of height and visibility of the experimental setting for each category. Thus, a small number means better accuracy. We test with ResNet-50 backbone network.

Setting	Height	Visibility
Reasonable	[50, inf]	[0.65, inf]
Small	[50, 75]	[0.65, inf]
Heavy	[50, inf]	[0.2, 0.65]
All	[20, inf]	[0.2, inf]

Table 3.3: Experiment Setting [1]

Table 3.4, presents results of existing CSP and revised with FcaNet one with Cityscapes dataset. FcaNet layered model achieves more accurate performance in this experimental setting. It shows improvement of accuracy by 0.49%, 0.55%, 1.8% and 0.67% . This shows the meaningful effectiveness of the FcaNet method.

	Reasonable	Small	Heavy	All
Existing	12.2	16.64	38.57	37.72
Revised	**11.71**	**16.09**	**36.77**	**37.05**

Table 3.4: Preliminary work result

3.2 Main Work

3.2.1 Architecture

We use three different models: Faster-RCNN, Libra-RCNN, and Double-Heads. We add the FcaNet layer to the convolution layer before each model. After equipping the FcaNet layer, it passes feature maps or other revised methods by each model. The architecture of each model is illustrated in Figures 3.5 - 3.7.

Every CNN has a convolutional layer that transforms the input image to extract features from it. In this transformation, the image is convolved with a kernel. It is the main building block of a CNN containing a set of filters or kernels, parameters which are to be learned throughout the training. the FcaNet layer initializes the DCT weights. It runs only at the very beginning and does not participate in the training and test. We use the same 16 frequency components as the preliminary work.

In Faster-RCNN and Double-Heads, the next step is to make feature maps, the result of applying the filters to an input image. At each layer, the feature map is the output of that layer. Due to the multi spectral attention of the FcaNet layer, the revised model has different outputs than the existing one. Libra-RCNN passes IoU Balancing and Balancing pyramid stage to match the portion of negative samples and integrate multi-level features using lateral connections.

The last step is RPN for Faster-RCNN, Softmax and Balanced L1 for Libra-RCNN and dividing head for Double-Heads. In RPN, a small network is a slide over a convolutional feature map that is the output by the last convolutional layer. RPN generates the proposal for the objects. RPN has a specialized and unique architecture in itself. The balanced L1 part will be illustrated in the loss function part later. Double-Heads takes the advantage of each benefit of $fc-head$ and $conv-head$ in classification and bounding box regression, respectively.

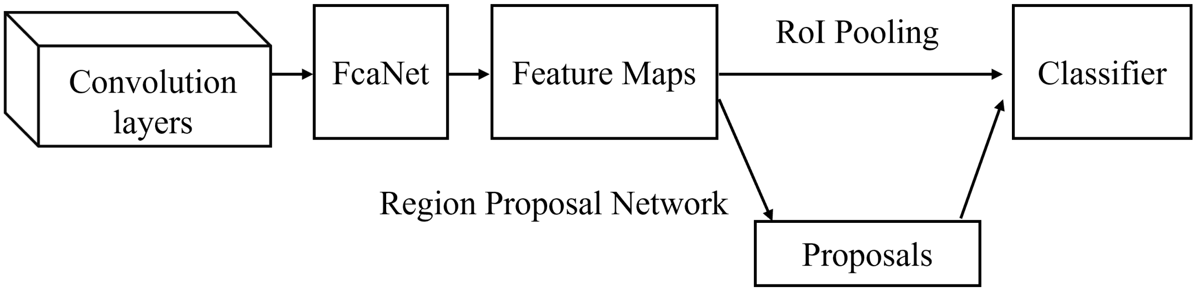

Figure 3.5: The overview FcaNet with Faster-RCNN

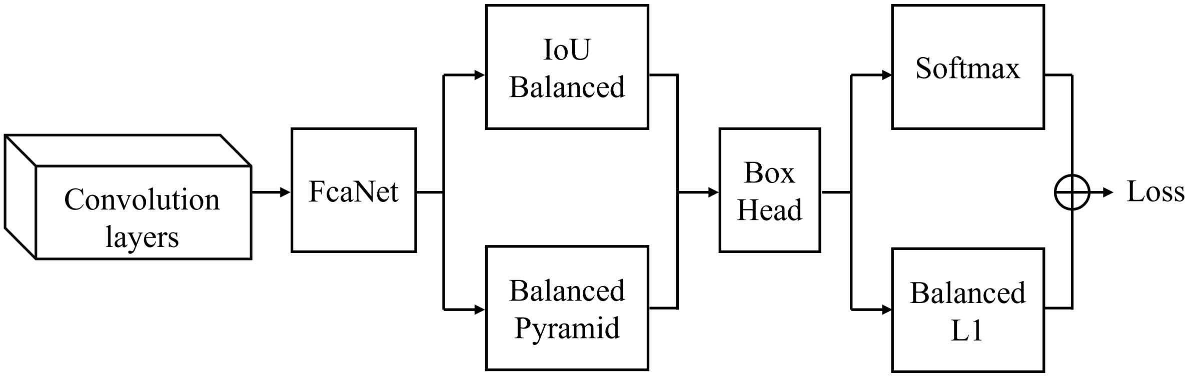

Figure 3.6: The overview FcaNet with Libra-RCNN

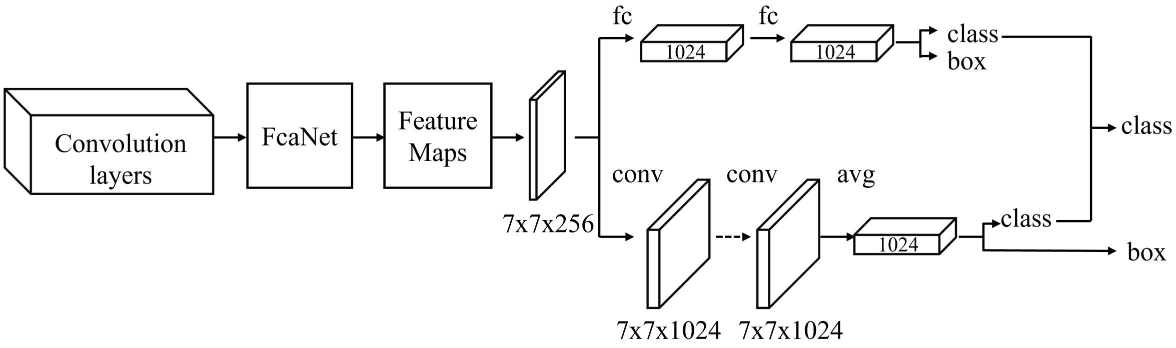

Figure 3.7: The overview FcaNet with Double-Heads

3.2.2 Using customized dataset

We collect various sizes and quality images to build construction site images, add unrelated ones to the working environment. Totally, 11,395 image data are collected, 9,116 and 2,279 images are assigned for training-validation and test. The annotation file is distinguished by 5: ’Person’, ’Head’, ’Helmet’, ’Jacket’, and ’Red-life-jacket’.

Figures 3.8 (a) - (g) show images and labels of the dataset. Basically, it covers ’Person’, ’Head’ when it comes to image of person. ‘Head’ is labeled for the head without a helmet, and ‘Helmet’ is labeled without a ‘Head’ if worn. In addition, a distinction is made between a general Jacket and a Red-life-jacket. In the case of the Red-life-jacket, it is classified because it is worn by workers exposed to a more dangerous environment, unlike a general jacket. In labeling, a distinction is needed in the not-wearing state. So, when wearing not PPE but a hat, it is labeled as ‘Head’. If the equipment is just lying around, a person is not wearing, or not wearing the right PPE, it is treated as nothing and is not labeled. Figures 3.8 (h) - (i) show some examples of these cases.

During this process, a problem is discovered in the CSP model with the dataset, which frustrates its use. In the case of the CSP model, it is a method of analysis based on the images and videos taken by the vehicle camera with a certain image pixel and size. Analyzing uniformly recorded images is the goal of constructing the dataset. However, it could not be used because it is incompatible with the images of various sizes used in this thesis. Therefore, we decide to use state-of-the-art models, which use the MSCOCO format dataset. We don not think we can judge what is good or bad here. This is because each data has its own goals and is executed accordingly. In the case of this thesis, the environment here can directly pass through the construction environment and shoot in a certain environment is inadequate. Therefore, there is a limitation in collecting uniform images. So, the models for the MSCOCO database, which can handle many types of sizes, fits the condition of this study.

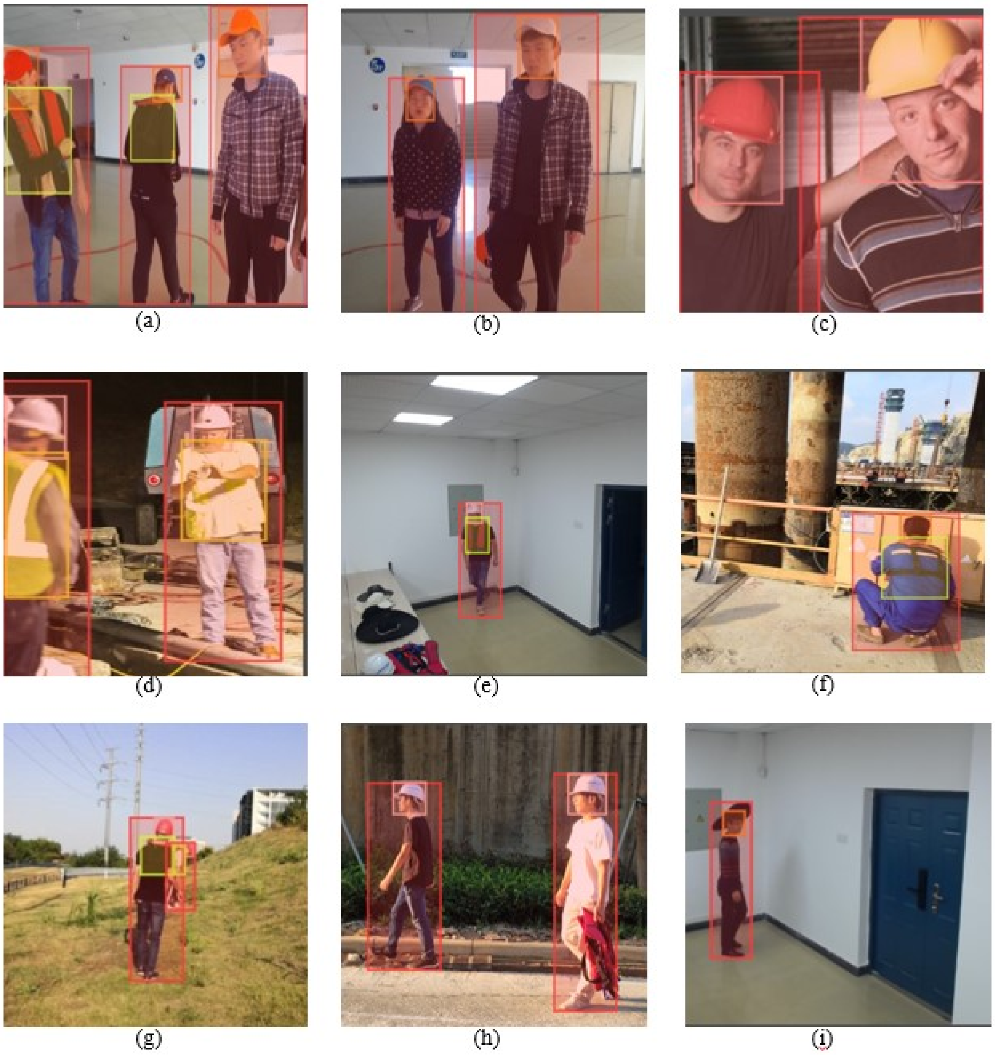

Figure 3.8: Labelling for each Image. (a) : Person, Head, Red-life-jacket (b) : Person, Head (c) : Person, Helmet (d) : Person, Helmet, Jacket (e) : Person, Helmet, Red-life-jacket (f) : Person, Head, Red-life-jacket (g) : Person, Helmet, Red-life-jacket (h) : Not labelling handled Red-life-jacket (i) : Not labelling non-helmet hats

However, 12k images have the following problems: the number of uniqueness and diversity is limited, challenging, occlusion samples are relatively rare. Since the target image itself is too small, there is a fatal flaw in improving accuracy or obtaining convergence when trying to get consistent test results. Therefore, an attempt is made to achieve the effect of pre-training using random initialization. GN/SyncBN are used to replace 'frozen BN'(channel-wise affine) layers, longer learning rate scheduling is conducted. We use '6×scheduling' which 540k iterations instead of 90k iterations in normal scheduling [14].

3.2.3 Loss function

The loss function of each model is different by revised equations. The loss function of the Faster-RCNN is defined as:

$$L(\{p_i\},\{t_i\}) = \frac{1}{N_{cls}}\sum_i L_{cls}(p_i,p_i^*) + \lambda\frac{1}{N_{reg}}\sum_i p_i^* L_{reg}(t_i,t_i^*) \tag{3.8}$$

where p_i is the predicted probability of anchor i being an object, and i is the index of an anchor in a mini-batch. If the anchor is positive, the ground-truth label p_i^* is 1, and if the anchor is negative, it is 0. The four parameterized coordinates of the expected bounding box are represented by the vector t_i. The ground-truth box associated with a positive anchor has a t_i^* value. The classification loss L_{cls} is a two-class log loss. $L_{reg}(t_i,t_i^*) = R(t_i - t_i^*)$ is the regression loss function, while R is the robust loss function. $p_i^* L_{reg}$ denotes that the regression loss is only active for positive anchors and is disabled otherwise. The cls and reg layers' outputs are $\{p_i\}$ and $\{t_i\}$, respectively. The two components are normalized by N_{cls} and N_{reg}, respectively, and weighted by λ, a balancing parameter [9].

Balanced L1 loss is developed from the usual smooth L1 loss in Libra-RCNN, and it promotes critical regression gradients from inliner to rebalance the involved samples and tasks, resulting in a more balanced classification training. The balanced L1 loss used by the localization loss L_{loc} is defined as [10]:

$$L_{loc} = \sum_{i\in(x,y,w,h)} L_b(t_i^u - v_i) \tag{3.9}$$

based on this, promoted gradient formulation can be written as:

$$\frac{\partial L_b}{\partial x} = \begin{cases} \alpha ln(b\,|x| + 1) & \text{if } |x| < 1 \\ \gamma & \text{otherwise} \end{cases} \tag{3.10}$$

$$L_b(x) = \begin{cases} \frac{\alpha}{b}(b\,|x| + 1)ln(b\,|x| + 1) - \alpha\,|x| & \text{if } |x| < 1 \\ \gamma\,|x| + C & \text{otherwise} \end{cases} \tag{3.11}$$

In Double-Heads, to use in unfocused task, it uses both loss functions. The overall loss is computed as follows:

$$L = w^{fc}L^{fc} + w^{conv}L^{conv} + L^{rpn}, \tag{3.12}$$

where w^{fc} and w^{conv} are weights for $fc - head$ and $conv - head$, respectively. L^{fc}, L^{conv}, L^{rpn} are the losses for $fc - head$, $fc - conv$, and RPN.

The loss for $fc{-}head$ for the unfocused task contains both classification loss and bounidng box regression loss, where L^{fc}_{cls}, L^{conv}_{cls}, and L^{fc}_{reg}, L^{conv}_{reg} are the $fc - head$ and $conv - head$ classification and bounding box regression losses, respectively. The weight that regulates the balance between the two losses is λ^{fc} [11].

$$L^{fc} = \lambda^{fc}L^{fc}_{cls} + (1 - \lambda^{fc})L^{fc}_{reg}, \tag{3.13}$$

$$L^{conv} = (1 - \lambda^{conv})L^{conv}_{cls} + \lambda^{conv}L^{conv}_{reg}, \tag{3.14}$$

3.2.4 Experiment setting

We use MMDetection and Google colaboratory (Colab). MMDetection decomposes the detection framework into individual components, making it simple to create a customized object detection framework. It provides high-performance support for most modern detection frameworks [19]. MMDetection is an open source object detection toolbox based on PyTorch. The toolbox includes mainstream detectors, alternative backbone networks, and detection necks.

Back-propagation and Stochastic Gradient Descent are used to train each model end-to-end. Each mini-batch is made up of several positive and negative image anchors that come from a single image. It is feasible to optimize for all anchor loss functions. We extract weights from a zero-mean Gaussian distribution with a standard deviation of 0.01 to randomly initialize all new layers.

The backbones are ResNet-50 and the FPN. The ResNet-50 network features a small number of layers, a rapid processing speed, and less excessive precision loss, allowing it to significantly reduce training time while maintaining high accuracy. The FPN can decrease feature loss in the training process and cope with the problem of target shape difference by applying multi-scale feature fusion. In each cycle, the training scale is chosen at random, and the picture is resized to fit the chosen scale. It pre-defines a scale range and generates a scale at random between the minimum and maximum values.

In Colab environment, we train with Pytorch framework and with one Tesla P100-PCIE-16GB GPU, mini-batch size of 2 images per GPU, NVCC: Build cuda11.1 version, TorchVision: 0.11.1+cu111, MMCV: 1.1.3. CuDNN 8.0.5 version. The weight decay is 1e-4 and momentum is 0.9. All models are tuned with 90k iterations except for using scratch method, the learning rate is initialized to 0.01.

To evaluate experiments on COCO format images, it is implemented on MMDetection within 70 epochs showing convergence. The standard COCO-style Average Precision(AP) with different IoU thresholds as evaluation metric. We use AP, $\text{AP}_{50}, \text{AP}_{75}, \text{AP}_{S}, \text{AP}_{M}$, and AP_{L} to show the performance of the improved network on multiple scales.

3.2.5 Main results

We compare Faster-RCNN with the FcaNet layer with the state-of-the-art object detection models on the customized dataset in Table 3.5. Faster-RCNN with the FcaNet layer achieves 37%, which is 5.3% higher AP than Faster-RCNN without FcaNet layer. Also, it shows 5%, 3.3%, 2.1% and 0.9% higher in AP than Empirical Attention, Re2net, Carafe and SENet[1], respectively.

Models	AP	AP50	AP75	AP_S	AP_M	AP_L
Faster-RCNN	31.7	65.6	27.4	5.7	25.3	33.9
Empirical Attention	32	66.3	27.7	7.7	25.9	34.1
Re2net	33.7	67.7	30.7	6.3	26.2	36
Carafe	34.9	69.4	32.4	7.2	29.2	36.9
Faster-RCNN (SENet)	36.1	70.6	33.9	7.6	30.3	37.2
Faster-RCNN (FcaNet)	**37**	**72.1**	**34.2**	**8.2**	**31.2**	**39**

Table 3.5: Results of FcaNet layer and the existing models

[1] The author would like to sincerely thank Prof. Serap Savari for suggesting the inclusion of SENet results for comparisons.

We further extend our FcaNet layer into other models and get the results shown in Table 3.6. Using Libra-RCNN, Double Heads, it achieves 36.7% and 36.2%, which is 0.4% higher in AP compared to the original one.

Models	FcaNet	AP	AP50	AP75	AP$_S$	AP$_M$	AP$_L$
Faster-RCNN		31.75	65.6	27.4	5.7	25.3	33.9
	○	**37**	**72.1**	**34.2**	**8.2**	**31.2**	**39**
Libra-RCNN		36.3	71.4	33.5	6.2	30	38.4
	○	**36.7**	**71.9**	**34**	**6.5**	**31.2**	**39.5**
Double-Heads		35.8	68	33.7	6.1	27.9	38
	○	**36.2**	**68.6**	**34**	**7.3**	**28.5**	**38.3**

Table 3.6: Results of the original and revised models

Using scratch dataset to enhance AP to cover limited dataset, we get 0.4%, 0.2%, and 0.4% higher for each model in AP. The results are shown in Table 3.7.

Models	FcaNet	AP	AP50	AP75	AP$_S$	AP$_M$	AP$_L$
Faster-RCNN		55.8	86.2	59.4	21	50.2	58.4
	○	**56.2**	**86.5**	**60.5**	**22.2**	**50.5**	**58.9**
Libra-RCNN		57.3	88.5	61.3	22.5	52.3	60
	○	**57.5**	**88.8**	**61.7**	**22.7**	**52.5**	**61.2**
Double-Heads		56.5	88.5	60.2	21.5	51.4	59.5
	○	**56.9**	**88.9**	**60.5**	**22.3**	**51.6**	**59.6**

Table 3.7: Results of the original and revised models with scratch

4. DISCUSSIONS AND CONCLUSION

4.1 Recommendation and limitation

4.1.1 Compatibility of FcaNet

The biggest advantage of the FcaNet layer gained through this research is that it does not significantly hinder the computation and is compatible with various models. The reason for not burdening is that there is no big difference in the calculation method between the FcaNet and the GAP. The features are reallocated in the frequency domain channel, so only the frequency domain transformation is introduced in the process. No additional parameters are introduced, so the amount of computation is not changed. Also, as we have seen, it can be used in various models. It shows additional accuracy improvements over those that have already been revised in other ways. We believe this is possible because it improves accuracy from using only the DC coefficient to using both the DC and additional AC coefficients in the DCT domain.

4.1.2 No sizable improvement except for Faster-RCNN

What we have observed through this work is that using multiple tools for improvement does not add all the benefits. In the case of the CSP model in the preliminary work, and Libra-RCNN and Double-Heads excluding Faster-RCNN in this thesis, these get improvements in accuracy by capturing and complementing the imbalance and missed part of the existing models as the FcaNet. Even if the complementary measures are repeated, each effect can not be fully achieved. This is same with not significantly increasing accuracy with multiple deep learning models.

4.1.3 Limitation of datasets

The most essential limitation of this data is that it only distinguishes between the presence and absence of equipment and lacks a judgment as to how it is. The models trained by our image dataset only recognize PPE and can not say whether it is properly worn by the user, that is, whether workers are guaranteed to be safe. It is possible to recognize that a person is wearing a helmet but can not tell if it is exactly worn on head. Therefore, it can be said that this research has achieved the effect on the most basic part for ensuring construction safety. To reach the discrimination through detection and recognition, data containing more situations must be added. There are many cases that need to be included to proceed to this step. In other words, we have to put in everything that can be happened while we are on the construction site. Not just wearing a helmet, it's just on it, upside down, distorted or not, tightly fastened chin strap, and so on.

Also, visual occlusion is the most common and hardest to solve problem [6, 7]. When a worker is partially or completely obscured by some objects, it can not be detected and monitored. Environmental factors also play roles that can not be ignored. For example, blocking the view from sunlight makes it difficult for the object detection models to monitor easily. Detection from a distance due to the physical positioning also contributes to reduced accuracy. There are some proposed solutions to deal with this problem: adjusting the camera positions, placing multiple cameras at the construction site. This helps to reduce some monitoring blind spots. Figure 4.1 shows the examples of elements diminishing accuracy. However, even the best algorithms still can not detect some occluded entities accurately due to the constraints of technologies, it should be one of the future research directions.

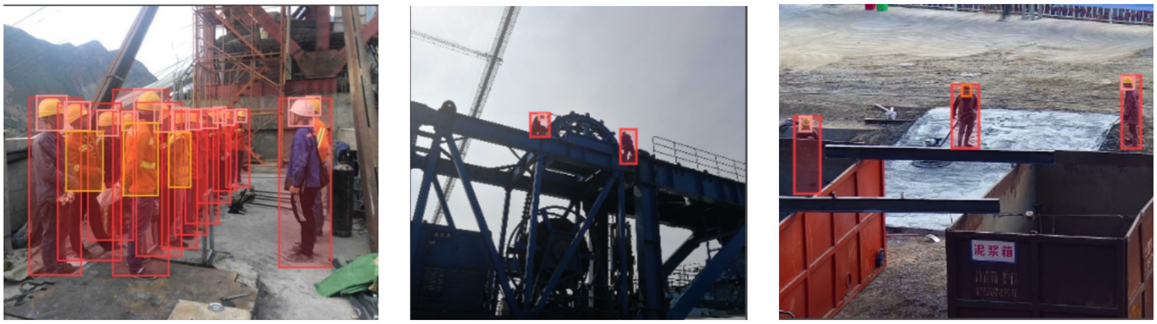

Figure 4.1: Reduced accuracy by crowded workers, far distance and occlusion

4.2 Conclusion

In this thesis, we use construction site images familiar with the real work environment. We have shown that the FcaNet with the multi-spectral attention module, which generalizes the existing channel attention mechanism in the frequency domain outperforms the existing models with the same number of parameters and computational cost. Also, it shows meaningful improvement and compatibility with many state-of-the-art object detection models. We have found the challenges and way forward of the application of the FcaNet layer in models: lack of all possible images, environment, and occlusion in dataset, limit to the usefulness of the FcaNet layer in that there is no significant improvement in accuracy on once revised models.

It is expected that this thesis will not only enhance understanding of the use of the classical method DCT in object detection can make progress but also provide insights into the computer vision based safety and health management in practice.

REFERENCES

[1] M. Cordts, M. Omran, S. Ramos, T. Scharwächter, M. Enzweiler, R. Benenson, U. Franke, S. Roth, and B. Schiele, "The cityscapes dataset," in *CVPR Workshop on the Future of Datasets in Vision*, vol. 2, 2015.

[2] R. Sehsah, A.-H. El-Gilany, and A. M. Ibrahim, "Personal protective equipment (ppe) use and its relation to accidents among construction workers," *La Medicina del lavoro*, vol. 111, no. 4, p. 285, 2020.

[3] M.-W. Park, N. Elsafty, and Z. Zhu, "Hardhat-wearing detection for enhancing on-site safety of construction workers," *Journal of Construction Engineering and Management*, vol. 141, no. 9, p. 04015024, 2015.

[4] N. D. Nath and A. H. Behzadan, "Deep convolutional networks for construction object detection under different visual conditions," *Frontiers in Built Environment*, vol. 6, p. 97, 2020.

[5] M. Schia, B. Trollsås, H. Fyhn, and O. Lædre, "The introduction of ai in the construction industry and its impact on human behavior," pp. 903–914, 07 2019.

[6] W. Liu, Q. Meng, Z. Li, and X. Hu, "Applications of computer vision in monitoring the unsafe behavior of construction workers: Current status and challenges," *Buildings*, vol. 11, no. 9, p. 409, 2021.

[7] Q. Fang, H. Li, X. Luo, L. Ding, H. Luo, T. M. Rose, and W. An, "Detecting non-hardhat-use by a deep learning method from far-field surveillance videos," *Automation in Construction*, vol. 85, pp. 1–9, 2018.

[8] Z. Qin, P. Zhang, F. Wu, and X. Li, "Fcanet: Frequency channel attention networks," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 783–792, 2021.

[9] S. Ren, K. He, R. Girshick, and J. Sun, “Faster r-cnn: Towards real-time object detection with region proposal networks,” *Advances in neural information processing systems*, vol. 28, 2015.

[10] J. Pang, K. Chen, J. Shi, H. Feng, W. Ouyang, and D. Lin, “Libra r-cnn: Towards balanced learning for object detection,” in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 821–830, 2019.

[11] Y. Wu, Y. Chen, L. Yuan, Z. Liu, L. Wang, H. Li, and Y. Fu, “Rethinking classification and localization for object detection,” in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 10186–10195, 2020.

[12] J. Hu, L. Shen, and G. Sun, “Squeeze-and-excitation networks,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 7132–7141, 2018.

[13] N. Ahmed, T. Natarajan, and K. R. Rao, “Discrete cosine transform,” *IEEE transactions on Computers*, vol. 100, no. 1, pp. 90–93, 1974.

[14] K. He, R. Girshick, and P. Dollár, “Rethinking imagenet pre-training,” in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 4918–4927, 2019.

[15] Y. Wu and K. He, “Group normalization,” in *Proceedings of the European conference on computer vision (ECCV)*, pp. 3–19, 2018.

[16] C. Peng, T. Xiao, Z. Li, Y. Jiang, X. Zhang, K. Jia, G. Yu, and J. Sun, “Megdet: A large mini-batch object detector,” in *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pp. 6181–6189, 2018.

[17] W. Liu, S. Liao, W. Ren, W. Hu, and Y. Yu, “High-level semantic feature detection: A new perspective for pedestrian detection,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 5187–5196, 2019.

[18] I. Hasan, S. Liao, J. Li, S. U. Akram, and L. Shao, “Generalizable pedestrian detection: The elephant in the room,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 11328–11337, 2021.

[19] K. Chen, J. Wang, J. Pang, Y. Cao, Y. Xiong, X. Li, S. Sun, W. Feng, Z. Liu, J. Xu, Z. Zhang, D. Cheng, C. Zhu, T. Cheng, Q. Zhao, B. Li, X. Lu, R. Zhu, Y. Wu, J. Dai, J. Wang, J. Shi, W. Ouyang, C. C. Loy, and D. Lin, "MMDetection: Open mmlab detection toolbox and benchmark," *arXiv preprint arXiv:1906.07155*, 2019.