QU-Net: A Lightweight U-Net based Region Proposal System

by

Rahul Santhosh Kumar Varma

A Thesis Presented in Partial Fulfillment
of the Requirements for the Degree
Master of Science

Approved November 2021 by the
Graduate Supervisory Committee:

Yezhou Yang, Chair
Deliang Fan
Yingzhen Yang

ARIZONA STATE UNIVERSITY

December 2021

ABSTRACT

In recent years, there has been significant progress in deep learning and computer vision, with many models proposed that have achieved state-of-art results on various image recognition tasks. However, to explore the full potential of the advances in this field, there is an urgent need to push the processing of deep networks from the cloud to edge devices. Unfortunately, many deep learning models cannot be efficiently implemented on edge devices as these devices are severely resource-constrained. In this thesis, I present QU-Net, a lightweight binary segmentation model based on the U-Net architecture. Traditionally, neural networks consider the entire image to be significant. However, in real-world scenarios, many regions in an image do not contain any objects of significance. These regions can be removed from the original input allowing a network to focus on the relevant regions and thus reduce computational costs. QU-Net proposes the salient regions (binary mask) that the deeper models can use as the input. Experiments show that QU-Net helped achieve a computational reduction of 25% on the Microsoft Common Objects in Context (MS COCO) dataset and 57% on the Cityscapes dataset. Moreover, QU-Net is a generalizable model that outperforms other similar works, such as Dynamic Convolutions.

ACKNOWLEDGMENTS

I would like to express my gratitude to Dr. Yezhou Yang for his continued guidance and support during the project. His motivation and vision have inspired me, and I appreciate the opportunity to work with him.

I want to thank my committee members, Dr. Deliang Fan and Dr. Yingzhen Yang, for their continuous guidance and support during the research work and the thesis.

I want to thank Mohammad Farhadi, who introduced me to the ARGOS and supported me at various stages of my work. Working with him has been a constant learning experience for me.

I also want to thank Active Perception Group as a whole. Although I was introduced to the in-person experience of the lab during the second half of my Masters, it was a great experience to work with other like-minded individuals - Varun, Prabal, Ashish, and Himanshu.

I would also like to thank my parents, Santhosh Kumar and Preetha Varma, and my fiancée Devika Unnikrishnan for everything.

TABLE OF CONTENTS

Page

APPENDIX

LIST OF TABLES

LIST OF FIGURES

Chapter 1

INTRODUCTION

1.1 Overview

Deep learning and neural networks have made great strides over the past decade with diverse applications in various fields of life. However, the inherent concept of neural networks is not new, with the McCulloch-Pitts Neuron model (1943) being the first mathematical model presented in the area of neural networks. With the rise of computers in the latter half of the 1900s, there were efforts to model the concept of a biological neuron and use these to develop an artificial neural network. However, the growth of neural networks was severely curtailed by the limitations in electronics. Neural networks and deep learning were catapulted back into the scene when AlexNet won the ImageNet challenge, the first Convolutional Neural Network (CNN) to win the competition. The success of AlexNet was propelled by the use of Graphical Processing Units (GPU) to accelerate the workload of neural networks.

Deep learning has been used for different applications in recent years, including computer vision and natural language processing. Neural networks have used more layers to learn latent features and achieve state-of-the-art performance. Consequently, this has led to an increase in the computational cost of these networks. As a result, there has been an increased focus on the computing power offered by GPUs to achieve the heavy computation required by deep learning-based applications. Propelled by the rise of AI and deep learning, the growth of GPUs has surpassed the expectations set by Moore's law - the number of transistors in dense integrated circuit doubles every two years - and has in turn been replaced by Huang's law - the speed at which

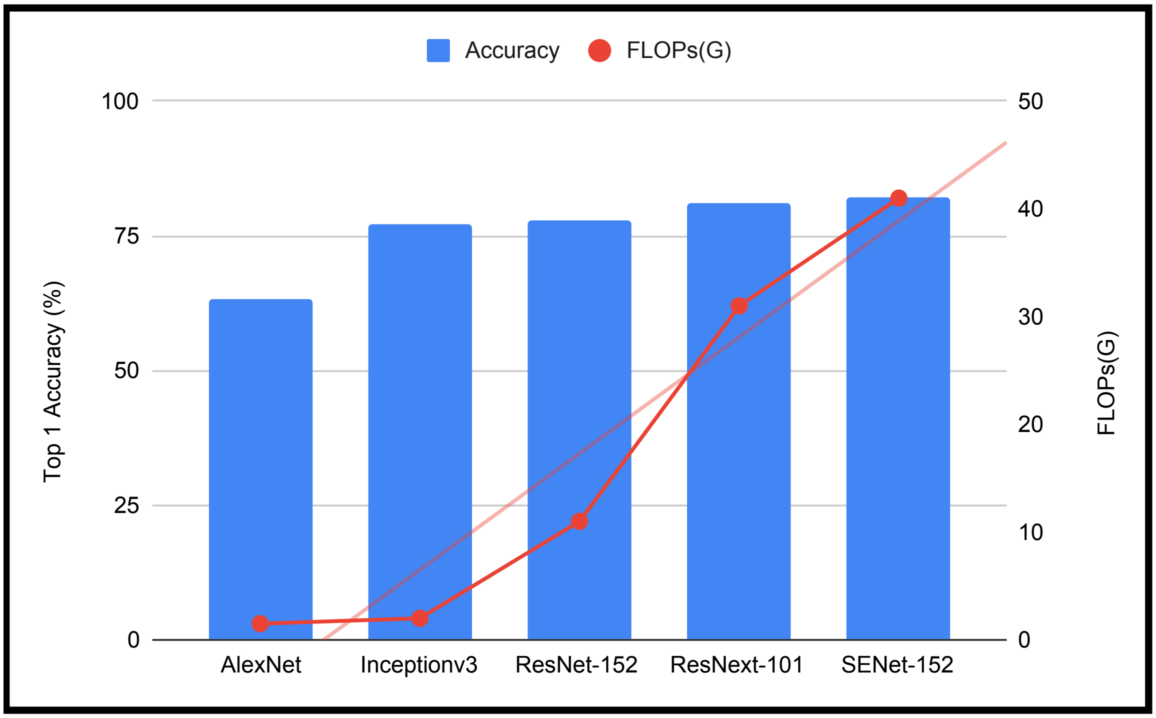
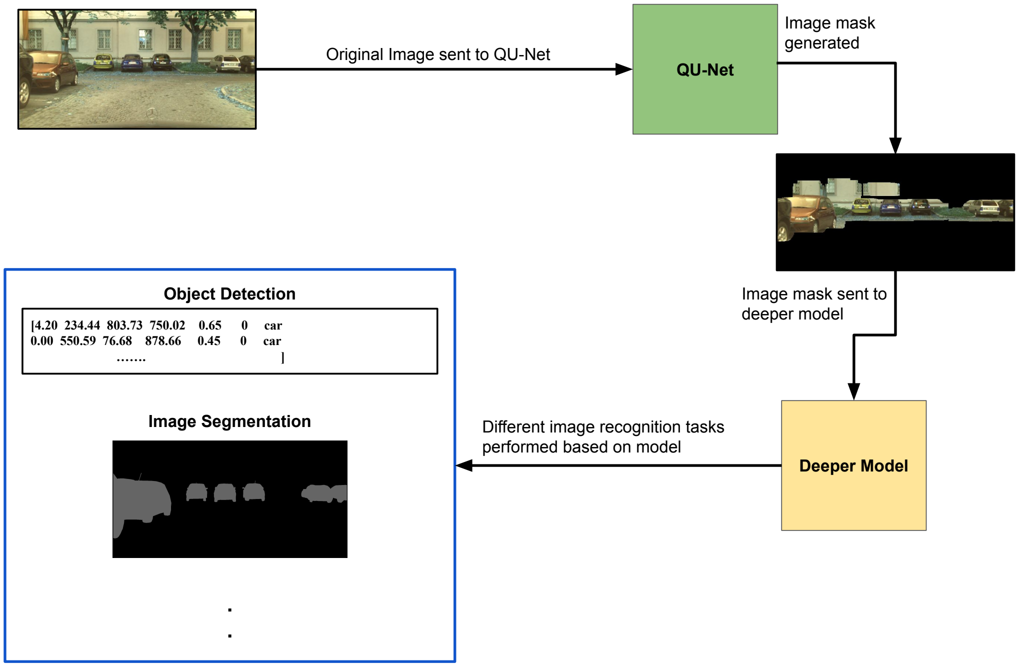


Figure 1.1: Top-1 accuracy Vs the amount of FLOPs for a model

GPUs grow is at a faster rate than that predicted by Gordon Moore. However, with the exponential increase in the complexity of deep learning models as shown in Fig. 1.1, there have been questions about the sustainability of deep learning in the long run. The improvement in accuracy does not grow at the same rate as the computational complexity, with a significant increase in computation required to achieve a minor increase in performance. Thus, it is crucial to develop more efficient and sustainable methods to train and use models. In addition, with the high computation requirements of these models, the implementation of the models has been restricted to cloud-based servers with powerful processors. Various devices with resource constraints currently send their data to a shared cloud server that processes the information and sends it back. This approach is unviable in the long run for two reasons, 1) The edge devices can generate a large amount of data that is received and processed by one central server. With the data projected to grow in the coming years, the cloud servers can be overloaded, disrupting an extensive network of devices. 2) The data being sent to the cloud cannot be fully assumed to be secure since the data is uploaded over a vulnerable network to the cloud.

1.2 Motivation


Original Image sent to QU-Net
QU-Net
Image mask generated
Image mask sent to deeper model
Deeper Model
Different image recognition tasks performed based on model
Object Detection
Image Segmentation
.
.

```
[4.20  234.44  803.73  750.02   0.65   0   car
0.00  550.59  76.68   878.66   0.45   0   car
                .......                        ]
```

Figure 1.2: The overall flow of the QU-Net-based baseline models

Most neural networks give equal importance to every region in the image. Many of these regions do not contain any objects, which does not necessitate extracting features and applying computation on these parts. The findings based on real-world data show that the total area occupied by the objects of interest in an image is much lesser than the background. For example, the Cityscapes Cordts *et al.* (2016) dataset, an example of a real-world urban dataset, have objects that only occupy an average area of 7% while the objects in the COCO Lin *et al.* (2015) dataset, a large dataset representing multiple everyday objects, occupies 34%. Based on these findings, I propose a lightweight model - QU-Net - that can efficiently predict the regions of interest and allow deeper models to focus their computation on these specific regions. QU-Net generates a binary mask representing the areas containing the object. The deeper model uses this masked image to direct their attention to those specific regions. The use of a masked input is theoretically equal to reducing the image size of the input. Nevertheless, to achieve a practical reduction, the convolution kernels need to be rewritten. The overall flow of QU-Net is shown in Fig. 1.2.

1.3 Challenges

While designing QU-Net, it was essential to consider the trade-off between the accuracy and the computation reduction with the question of how much an accuracy degradation was acceptable for an associated computation reduction. Although our model is used as a preprocessing step by other deep models, it is vital to maintain a good accuracy while determining the regions of interest so that the deeper model does not miss them. Building a quantized network involves an unavoidable accuracy degradation, especially for complex image recognition tasks like instance segmentation, although it provides the benefits of a lower computational cost. The challenge was to develop a model capable of detecting the maximum number of object regions, easily integrable with different deeper models, and achieving a sound computation reduction with an acceptable accuracy degradation factor.

1.4 Contribution

As part of this work, I aim to build a generalized region proposal model that can be used by models supporting different image recognition tasks. QU-Net reduces the computation of deep models by predicting the regions of interest on which the network should focus. While many works have explored the effect of quantization on fully convolutional networks, very few have explored the quantization of encoder-decoder architectures. The principles that can be followed for the traditional CNNs while quantizing may not apply to the encoder-decoder model. Further, binary neural networks have primarily focused on image classification datasets. The performance of these networks on more complex image recognition tasks has not been studied much. The main contributions of this work are -

- An extensive research on the effect of quantization on different parts of the U-Net network and the utilization of a DCT transformed image as an input.
- A binary segmentation model architecture called QU-Net that predicts the regions of interest, which are the regions with the highest probability of containing an object. QU-Net is proposed as a preprocessing step for deeper models to determine the regions suitable for processing.
- Experiments to evaluate the model's overall performance when implemented on different model architectures such as convolutional neural networks and transformers and another similar method (Dynamic Convolutions) that utilizes the spatial information to reduce the overall computation of a network.

1.5 Outline

- In **Chapter 2**, I provide an overview of region proposal systems and binary neural networks. Further, I provide a historical review of the various concepts in designing QU-Net, including region proposal systems, image segmentation, quantized neural networks, and dynamic models.
- In **Chapter 3**, I introduce and describe the approach in developing the QU-Net region proposal model, which includes the datasets used for training and evaluation, the architecture, and the overall training procedure.

- In **Chapter 4**, I talk about the different experimental setups and the corresponding results. First, I describe the effect of quantization on the different parts of the model. Further, I compare the work with Dynamic Convolutions, which is closest to this work. I also compare the work with several state-of-the-art methods to understand the overall performance of the network. Finally, I talk about the effect of different object sizes on the accuracy of the model.

- In **Chapter 5**, I talk about ARGOS, an end-to-end framework for accelerating neural networks on devices that have limited computation power, and how QU-Net integrates into the entire ARGOS architecture.

- In **Chapter 6**, I conclude the thesis and give a summary of the contributions of this work as well the potential avenues for future research in this area.

Chapter 2

BACKGROUND AND RELATED WORK

2.1 Background

2.1.1 Region Proposal

Object Detection models have commonly followed either a two-shot architecture or a single-shot architecture. A two-shot architecture usually consists of a region proposal stage responsible for defining the relevant regions of interest, and a classification stage categorizes the regions and refines the predictions. In contrast, a single-shot architecture does not consist of a region proposal stage and performs classification and localization in one step.

R-CNN Girshick *et al.* (2014) introduced the concept of region proposal where a selective search algorithm was used to propose an initial region of 2000, which was fed into a network individually to determine their classifications. The R-CNN method was slow since each image had to process 2000 regions that may or may not contain relevant classes. Later, faster variants such as and Fast R-CNN Girshick (2015) and Faster R-CNN Ren *et al.* (2016) were developed. Fast R-CNN proposed to alleviate this problem by using the feature map generated by the convolutional neural network to propose the regions. Fast R-CNN reduced the computation time since the convolution operation only had to be performed once per image. Both R-CNN and Fast R-CNN used selective search to propose the regions which Faster R-CNN aimed to replace by using a separate network to determine the regions from the feature map output. Faster R-CNN improved the overall run-time significantly and could be run real-time as well.

SSD Liu *et al.* (2016) YOLO Redmon *et al.* (2016) are two popular single-shot approaches. These architectures do not rely on regions and perform the detections in one step. One-Shot architectures have a lower accuracy when compared to two-shot networks, although they have a low run-time, while two-shot architectures yield high accuracy but have a higher inference time. The main cost in two-shot architectures is induced from the cost of the region proposal system. There has been ongoing work to improve the accuracy of single-shot networks to match that of two-shot architectures.

A model always has a lower computational complexity on smaller images because the total area to be processed is much smaller. Therefore, if we were to have a network that predicts the image locations, it would help reduce the image's total area, similar to a region proposal system. To ensure the cost of the region proposal model was small, we used a quantized network. Further, both single-shot and two-shot architectures can be used as an image area reduction benefit both networks in reducing the computation.

2.1.2 Binary Neural Networks

Before I dive into the related work and the proposed approach, it is essential to understand the concepts behind binary neural networks, which is the building block of this work. Binary neural networks are not a recent concept, but they have gained popularity in recent years. The core idea behind this is using a lower bit representation for the weights and activations in the network that can benefit both memory and computational complexity.

Binary neural networks employ a function to convert all the tensor values to +1 and -1. The binarization of values drastically reduces the memory consumption as the numbers need only 1-bit for storage rather than 32 bits. The sign function is shown in 2.1 which assigns every value greater than 0 to +1 and -1 otherwise. Here,

x_b refers to the binary value of the real number.

$$x_b = \begin{cases} +1 & \text{if } x \geq 0 \\ -1 & \text{otherwise.} \end{cases} \tag{2.1}$$

One issue associated with binarizing the values is that it loses its continuity and becomes a discrete function. This makes the function non-differentiable during the backpropagation phase while training the network. A widely used solution in the training of binary neural networks is the straight-through estimator (STE), as shown in Fig. 2.1 [1]. The STE method passes the gradients of the previous step to the next step for a binary layer. It has been shown to work very well and helps circumvent the issue brought forward by the sign function.

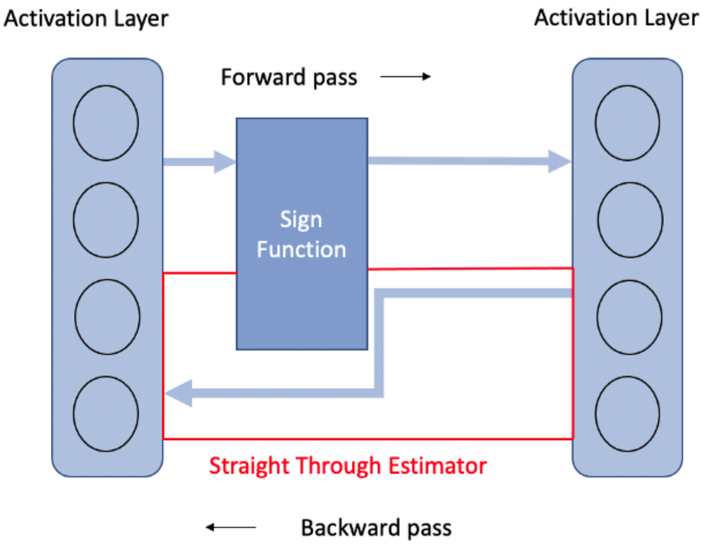


Figure 2.1: Straight Through Estimator approximation during backward pass

At each gradient accumulation phase, the binary weights and activations are used

[1] Own image - https://towardsdatascience.com/binary-neural-networks-future-of-low-cost-neural-networks-bcc926888f3f

to calculate the loss. However, the loss is backpropagated such that the weight update happens on the full-precision values. Once all the parameters are updated, they are clipped between -1 and +1.

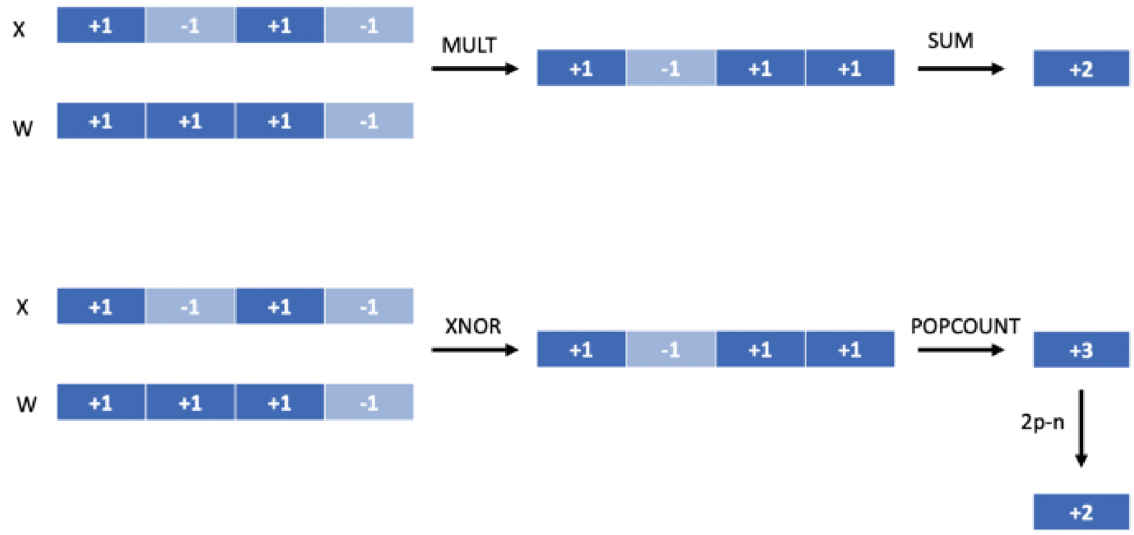


Figure 2.2: Bitwise XNOR operation as a replacement to convolution operations

Binary neural networks can also give significant improvements in terms of computation. XNOR-Net Rastegari *et al.* (2016) introduced a bit-wise operation that can be used to replace the multiply-accumulate operation. It has been shown to achieve a significant speedup because bit operations are internally implemented in various hardware devices, processing it faster than other arithmetic operations. A combination of XNOR and popcount can replace the convolution operation as shown in Fig. 2.2 [2]. The XNOR operation outputs a one if both inputs are the same, otherwise 0. The popcount operation is used to count the number of bits equal to 1. The illustration shows how an XNOR+popcount operation can be used to replace the multiply-accumulate operation. It shows that when the inputs are constrained to +1

[2] Own image - https://towardsdatascience.com/binary-neural-networks-future-of-low-cost-neural-networks-bcc926888f3f

and -1, they produce the same output as normal arithmetic operations.

The reduced bit representation also results in accuracy degradation compared to full-precision networks because of the low representation capacity of the tensors. There has been ongoing research in this area, with various advancements made to improve the overall accuracy of these networks. The main reason for the loss of accuracy in binary neural networks for various complex image recognition tasks is the loss of local information on binarization. I aim to use a lightweight binary model for the task of region proposal, which does not depend a lot on the local information.

2.2 Related Work

2.2.1 Background Subtraction

Background subtraction is the task of separating the foreground from the background. It is mainly used to separate moving objects from the surrounding environment and is widely used as a preprocessing step for multiple image recognition tasks. There have been many methods proposed for background subtraction. However, the general idea is to use the information from successive frames to determine the motion vectors of each pixel and predict the static pixels, which can be classified as background.

Currently, various OpenCV techniques and algorithms provide efficient methods to separate the foreground from the background, which is the concept used to determine the regions of interest in the image. There have also been multiple background subtraction-based methods proposed, such as Nguyen and Choi (2020); Sengar and Mukhopadhyay (2020) that use this technique for processing relevant spatial regions. However, these methods find it difficult to adapt to the natural motion in the scene and the camera movement. To build a model that is not hindered by any of the irrelevant motions in the scene and can efficiently extract the foreground, I propose a general-purpose region proposal model that uses a segmentation mask to determine the object locations used as a region proposal output. There have been works in the past Lim and Yalim Keles (2018); Lim and Keles (2020) that have used the concept of segmentation for background subtraction, but these models are computationally heavy. At the same time, I aim to build a system that is a lightweight preprocessing model for deeper models. The proposed method can also predict specific classes of objects without focusing on all the regions, which is particularly useful in scenarios where only certain classes are of interest.

2.2.2 Image Segmentation

Image segmentation is an essential aspect of many visual understanding systems. Segmentation is the process of partitioning the image into multiple regions based on the different objects present in the scene. It can be as simple as classifying the image pixels into two classes of foreground and background or classifying each pixel as a separate class in a dataset of multiple classes. In recent years, deep learning-based models have been developed that have shown significant improvement over traditional models. Many segmentation schemes utilize the encoder-decoder model. The latent features are extracted in the encoder stage, which the decoder utilizes to create the final output. U-Net Ronneberger *et al.* (2015) is a widespread implementation of this type of network, which has been widely used in the biomedical field. The U-Net model is utilized as the baseline to create the binary segmentation model. Although U-Net has been mainly used in the biomedical sector, it extends well to various other scenarios. Building a quantized region proposal system using U-Net reduces the computation of deeper models using the masks generated.

In the past, papers such as Segnet Badrinarayanan *et al.* (2015) have proposed a light-weight segmentation model. Nevertheless, this model still requires more computation than the one proposed and finds it difficult to extend to datasets containing multiple classes. Further, works such as DeepLab Chen *et al.* (2017), and PSPNet Zhao *et al.* (2017) are more advanced works that have managed to achieve good accuracy. However, the cost of these networks is significantly higher than that of U-Net, which does not bring the same benefits on quantization as the U-net network. Since the main focus of this work is on building a region proposal system that does not depend highly on the accuracy of the segmentation labels, the lower-accuracy network can be utilized to achieve higher benefits.

2.2.3 Quantized Neural Networks

Deep neural networks contain a large number of layers with millions of parameters. These allow the networks to learn complex information, but it comes at the cost of high memory requirements and computational complexity. The high cost of deep neural networks is attributed to the high multiply-accumulate operations in the convolution layers, making it challenging to run many of these deep models on low-power devices. Therefore, there has been ongoing research to develop and implement deep neural networks on such devices in recent years. One method proposed to achieve this is by reducing the bit representation for both weights and activations. The aforementioned reduces the memory as well as the computation requirements for the entire network.

Some of the initial works which explored the concept of binary networks are Courbariaux *et al.* (2016); Rastegari *et al.* (2016). They proposed to achieve state-of-the-art results on the image classification tasks accompanied by a reduction in memory and computational cost. This speedup was achieved by binarizing the weights and activations in the different layers of the neural network. The binarization allowed the speedup of convolution operations which are the main bottlenecks of a network, by replacing them with XNOR+popcount operation. The gain achieved in terms of the computational complexity of the convolution operation was 64 times theoretically. The cost of a convolution operation, which is the number of operations it performs, is directly proportional to the n - the number of channels, N_W - the filter size and N_I - the input size. To improve the accuracy, XNOR-Net implements a channel-wise scaling factor that reduces the speedup by a slight factor but provides much higher accuracy. However, the accuracy achieved was still not close to that of the full-precision methods. Further, XNOR-Net also implemented a model based on binary weights without changing the precision of the activations, which managed to reach state-of-the-art accuracy. There has been ongoing research in this field, but very few networks seldom achieve the same accuracy in other image recognition tasks, with the difference being unreasonable. One reason for this is that the network loses much of the local information on quantization. This results in an accuracy degradation as both the local and global context are essential while making predictions. The proposed region-proposal model relies less on pixel-wise accuracy and more on estimating the object presence, decreasing the significance of the local descriptors. This allows us to overcome the shortcomings of quantized networks and use the less accurate local information to predict the object hotspots.

Traditionally, binary networks avoid the binarization of the first and the last layers since it hurt the accuracy significantly. Moreover, since the cost of a convolution operation is proportional to the filter size and number of channels, it did not achieve a high reduction in computation when binarized. However, many recent neural networks use high-resolution images with the image size contributing to the complexity of a convolution operation. This can cause the first layer to become a bottleneck, mainly when many images are being processed with the effect of achieving real-time inference. Therefore, the binarization of initial layers should be handled with care since they are the most critical layers during quantization. Any information lost due to reduced information capacity cannot be obtained in the later layers. They are crucial in extracting the initial features for various image recognition tasks and passing them to the deeper layers of the network. By the time information reaches the final layers of the network, the feature maps are well defined, which results in a lower information loss on binarizing these layers. Flexible binary networks Wang *et al.* (2018a) utilized this observation to binarize a network based on inverse depth priority. The binarization started from the last layer, with each iteration working towards the top. At each step, the current layer was binarized, and the lower layers adjusted to maintain the accuracy loss at a minimum.

Quantization of the different layers also provides great benefit, albeit to a lesser extent, compared to a binarization scheme. It is important to reduce the bitwidths of both weights and activations if we want to reduce memory and computational cost. The activations form a major part of the memory map, especially when we use a large number of batches. Since the proposed model was designed to be using different activation schemes, Zhou *et al.* (2016) introduced a method to train and deploy quantized models with different quantization schemes on various devices efficiently. DoReFa-Net also replaced the channel-wise scaling factor introduced by XNOR-Net with a constant scalar. This allowed the network to utilize the bitwidth kernels during backpropagation, allowing a speedup during training and inference. The paper also defines the convolution kernel, which can be used for low bitwidth fixed point integers as defined in Eq. 2.2. Here, $c_m(x)$ and $c_m(y)$ are the bit vectors. The complexity of a convolution kernel is directly proportional to the number of bits in x and y which are the weight and activation tensors.

$$r = \sum_{m=0}^{M-1} \sum_{n=0}^{N-1} 2^{m+n} bitcount[and(c_m(x)c_m(y))] \tag{2.2}$$

To ensure there was no accuracy degradation on quantizing the initial layers, we had to ensure the activations had enough representation capacity after reducing the bitwidth. Banner *et al.* (2018) has shown that the accuracy drop with the use of 8-bit representations is negligible when compared to full-precision networks. It was posited that the quantization of vectors is highly dependant on the distribution of vectors with a Gaussian distribution less susceptible to quantization. The weights and activations follow a Gaussian distribution which allows us to quantize them effectively. The gradients in a network do not follow the same distribution and are more fragile. Since our model aims to be used at inference time after offline training, we can safely use full precision for the gradients while quantizing the weights and the activations.

2.2.4 Dynamic Neural Networks

The ability of a neural network to adapt itself based on the output is the essence of dynamic neural networks Han *et al.* (2021). These can be classified into three main categories; Sample-wise dynamic models, Spatial-wise dynamic models, and Temporal-wise dynamic models. Sample-wise dynamic models are designed to allow dynamic architectures that can support different inputs. There are two main categories of dynamic architectures - dynamic architectures and dynamic parameters. The dynamic architecture approach is one where the architecture of the network is conditioned based on the complexity of the input. This includes approaches where a different number of layers are executed, the width of a network is dynamic or selecting an appropriate branch from a tree-like neural network (SuperNet). The dynamic parameters approach includes methods to perform a weight adjustment, predict the weights, or apply a soft-attention to the weights.

Temporal-wise dynamic models Yeung *et al.* (2017); Su and Grauman (2016) reduce computation by dynamically determining the computation required for each frame. This can include methods where the features from previous frames are reused, specific frames are skipped, and the network evaluates results based on a small part of the video. Both Sample-wise and Temporal-wise methods mentioned above have been proved to be effective in reducing the computation of the network. However, one major drawback is that they consider an entire image to fall into either one of the classes - hard or easy. In contrast, a single image can contain both hard-to-classify objects and ones that can be easily classified.

The spatial-wise dynamic model utilizes the fact that different regions in an image have different complexity in image recognition tasks. Spatially Adaptive Computation Time (SACT) Figurnov *et al.* (2017) is an example of sample-wise dynamic models which work on the above principle. It also selectively borrows from sample-wise dynamic execution by implementing a network that executes the deeper layers based on the complexity of the image. SACT is modeled using ResNet architecture as its building block. It uses an Adaptive Computation Time(ACT) to compute the point of stoppage for each input. ACT adds an additional branch to each residual block, predicting a score between [0,1] known as the halting score. The global score is maintained as the input proceeds through the network, stopping the computation of different residual blocks. SACT takes it one step forward by using ACT on different spatial blocks in the input. The different blocks in the input are evaluated with each position inactivated when it attains a halting score greater than one. The other spatial regions are evaluated until all the regions in the input complete execution. The SACT model is not flexible as it requires deep residual networks to be effective because it relies on feature refinement.

Dynamic Convolution Verelst and Tuytelaars (2020) is another work based on dynamic spatial complexity and is probably the closest to this work. Dynamic Convolution is also based on the ResNet network. Contrary to SACT, which creates a spatial mask at each residual block. This mask is used to process the regions at the next step selectively. The work also implemented its own gather-scatter approach that enabled a practical speedup of the operation. Although it has shown excellent results, it has limited adaptability since extending it to non-ResNet-based models is difficult. Moreover, the use of dynamic convolutions also requires modification to the original networks, with the need to add the additional capability to ResNet to predict the mask. This work, QU-Net, proposes a lightweight region proposal system that can fit any model as a plug-and-play system and be used on models that support different tasks such as object detection, instance segmentation, and pose estimation without a separate training cycle. Furthermore, no changes need to be implemented on the original deeper network, with the only change being the input fed into the deep neural network.

Chapter 3

PROPOSED METHOD - QU-NET

In recent years, binary neural networks have been proposed as a promising technique to train and deploy deep neural networks on resource-constrained devices. However, although binary neural networks help reduce memory and computation costs, they bring forth the loss of information because of the low representation capacity and the difficulty in optimizing the network during training because of the discontinuous band. There have been various methods proposed that solve the above issues. However, binary neural networks still have not achieved an accuracy comparable to full-precision models, especially in object detection and image segmentation.

This work proposes a binary neural network, QU-Net, a binary segmentation model adopted as a region proposal model. This model can be used as a preprocessing step for any deeper model to remove the regions from the input that do not contain any relevant objects. The deeper model is only required to process the regions of interest, thus reducing the overall computation of the network resulting from the smaller image area processed.

3.1 Datasets

Many binary models have been trained and tested on the ImageNet dataset for the image classification task. However, ImageNet does not completely encapsulate the real-world scenario since many images contain canonical perspectives and a single label. In addition, full precision models have evolved to perform more complex tasks such as multiclass object detection and instance segmentation in recent years. Compared to the full-precision models, the binary models have shown underwhelming performance in these categories, a significant hurdle affecting their widespread adoption. To ensure the model was applicable in real-world scenarios, complex datasets (MS COCO and Cityscapes) that mimicked the real-world scenarios were used to train the model and gauge its performance.

3.1.1 COCO

Microsoft Common Objects in Context (MS COCO) is a dataset created to advance the field of image recognition. It has been widely used to benchmark various SotA models. It consists of labels for object detection, segmentation, and pose estimation. It consists of annotated objects from more than 80 classes, including everyday objects such as car and person. The total number of images in the training and validation set are as follows -

- Train Set - 118287 images
- Validation set - 5000 images

Figure 3.1: Sample images from the COCO dataset

3.1.2 Cityscapes

Cityscapes is a dataset that consists of images from various urban street scenes. It consists of labels for image segmentation and depth estimation, with unofficially supported object detection labels. The images were captured from more than 50 cities in varying conditions to build a highly dynamic and varied dataset. A few sample images are shown in 3.2. There are 19 classes defined for evaluation which include car, person, and bike. It consists of the following number of images in the training and validation set -

- Train Set - 2975 images
- Validation set - 500 images

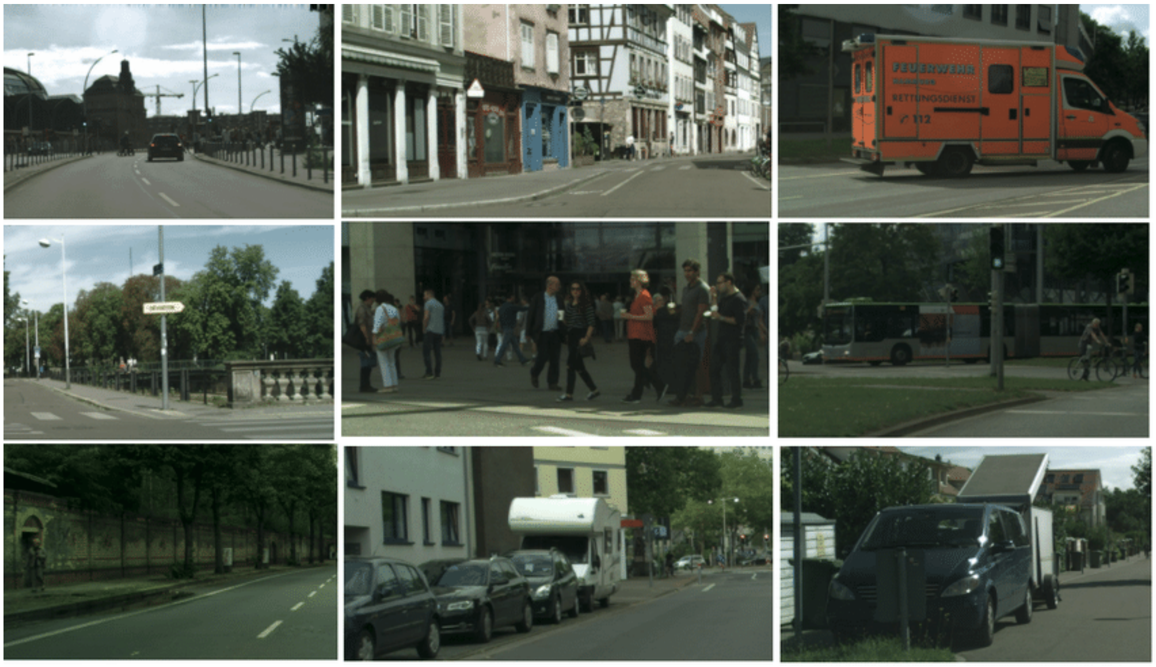


Figure 3.2: Sample images from the Cityscapes dataset

3.1.3 Differences between the two datasets

The COCO dataset is one of the most extensive publicly available datasets containing images of everyday objects. The representation of objects in their non-canonical perspectives sets COCO apart from ImageNet, which allows the model to learn general representations of the object. Moreover, COCO also consists of varied classes that will enable us to test the ability of the model to adapt to a large number of mixed classes. Although COCO consists of many classes, Cityscapes better represents real-world urban scenes and traffic images. Further, it also has a diverse set of data from multiple cities under different conditions. Thus, Cityscapes is essential to model deep neural networks that can be used in real-world scenarios, including self-driving cars, traffic cameras, and many more. These are the areas where the application of low-power neural networks is the most crucial.

3.2 Architecture

QU-Net was adapted from the U-Net Ronneberger *et al.* (2015) model. U-Net was initially developed as a replacement to fully convolutional networks for the image segmentation task in the biomedical field. Subsequently, U-Net has been used in other domains and evaluated on diverse datasets. U-Net also has a low computational cost when compared to more recent segmentation models. While developing a region proposal model, it was essential to build a low complexity model to ensure it did not add additional complexity to the deeper models that used QU-Net as a region proposal model, and the overall complexity was reduced. The weights across the entire network, except the last layer, were binarized. However, the model was divided into three parts for the activation layers, with each part following its own set of quantization rules.

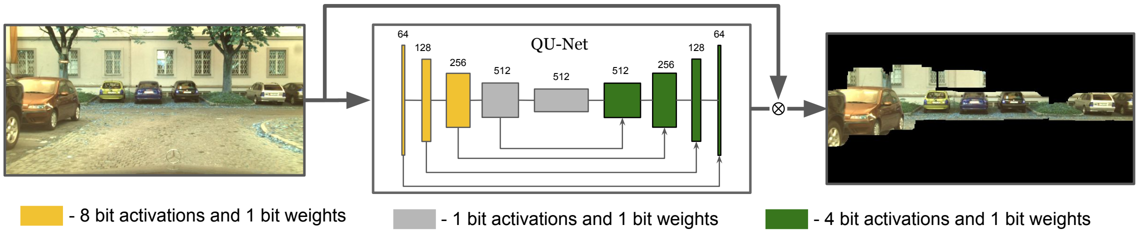

QU-Net
64
128
256
512
512
512
256
128
64
- 8 bit activations and 1 bit weights
- 1 bit activations and 1 bit weights
- 4 bit activations and 1 bit weights

Figure 3.3: QU-Net: A U-Net-based binary model that predicts the regions of interest in images that can be used for further processing

XNOR-Net Rastegari *et al.* (2016) proposed a binary-weight model that maintained the accuracy while reducing the memory requirements. Afterward, more works brought forward the fact that the accuracy of a model does not degrade when we binarize the weights exclusively. This observation was the primary motivation behind using a complete binary weights network. Moreover, it also provided significant advantages in terms of memory reduction. Experiments were also performed to ensure there was no accuracy degradation during the binarization of weights.

Initial experiments were performed to understand the importance of the different layers in the U-Net model and determine the effects of binarization. Previous works such as Wang *et al.* (2018a) have posited that as one goes deeper in a convolutional neural network, the effect of the quantization of layers on the accuracy reduces. This holds for a fully convolutional network, but very few works have explored the effect of quantization on encoder-decoder architectures. Like fully convolutional networks, the initial layers in the U-Net network are the most important because any information lost at these stages cannot be reclaimed at later stages. Binary neural networks avoid applying any quantization functions on the initial layers to prevent information loss. It was determined that the first three layers in the encoder network were important in feature extraction as there was significant accuracy degradation on binarizing these layers. Once the information is passed down from the initial layers to the middle layers, vital information has been extracted from the initial samples, allowing the flexibility to use a lower representation before the decoder stage. A set of 1-bit weights and 1-bit activations (represented by the gray regions in Fig. 3.3) were used in the final two convolution layers of the encoder part in the network.

Unlike typical fully connected networks where the final layers have lesser importance when compared to the initial layers during quantization, the decoder part of a U-Net model holds significance since it plays an essential role in reconstructing the final image mask from the extracted features. Thus, it is essential that the information capacity of the decoder part is high such that the information is passed up to the last layer with high accuracy. The decoder can be considered as a separate, fully convolutional network with an equivalent priority of layers. Experiments were performed with different activation bitwidths to understand the specific quantization at which the model maintains or loses marginal accuracy. The experiments showed the model maintained the accuracy when a bitwidth of the activations layer was reduced to 4 with a severe degradation in accuracy on using lower bits. This resulted in using a 4-bit representation of activations with a 1-bit representation of weights. The last layer was maintained as a full precision layer to allow the network to output a range of values.

To ensure the model was efficient, it was essential to quantize the initial layers as well. This was to ensure the initial layers did not act as a bottleneck. Research Banner *et al.* (2018); Wang *et al.* (2018b) has shown that deep neural networks using 8-bit representations can achieve the same accuracy as their 32-bit counterparts. Thus, a set of 8-bit activations and 1-bit weights were used in the initial convolution layer and the first two downsampling layers (represented by the yellow regions in Fig. 3.3). Also, using a lower quantization scheme hurt the accuracy badly. Compared to the encoder layer, the initial layer of the network requires a higher quantized representation (8-bit) which can be attributed to the fact that the effort required to extracting the initial features from the entire image is more than rebuilding a binary mask from the extracted features.

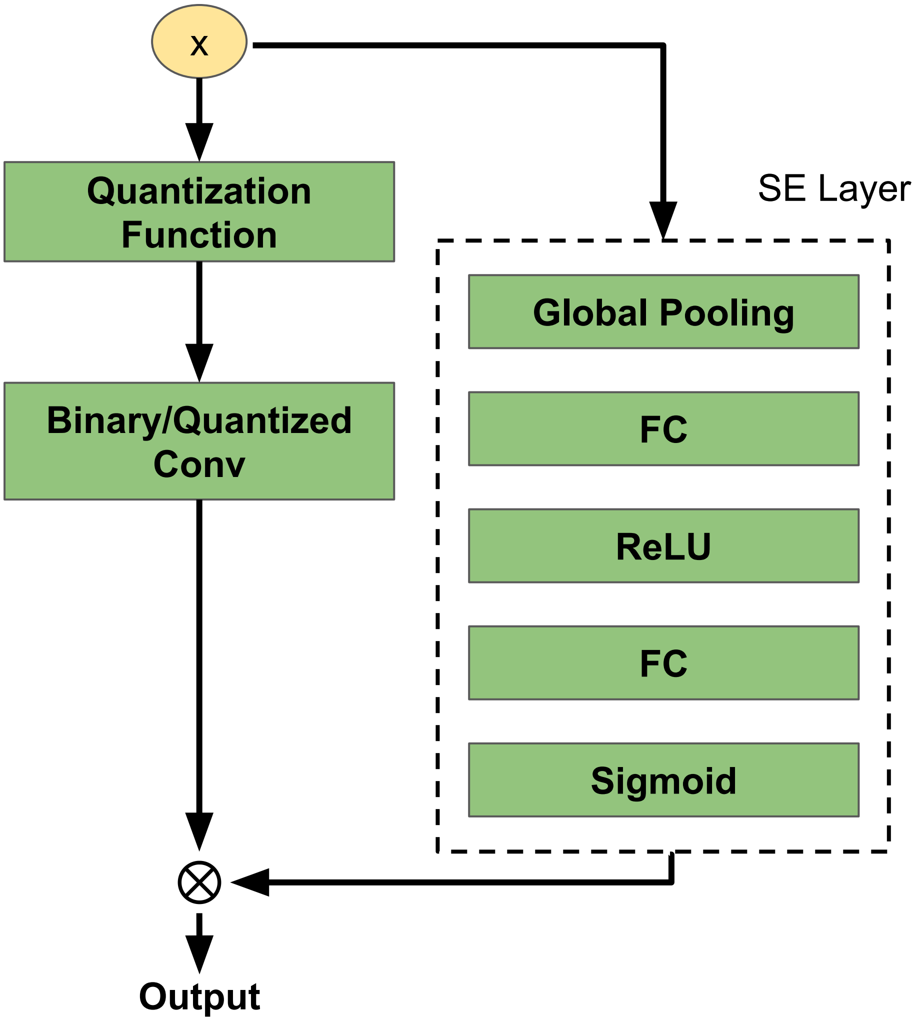


Figure 3.4: Architecture Diagram of the Quantized Convolution Layer Implemented

Each quantization module had a squeeze-and-excitation (SE) block Hu *et al.* (2018)it. An SE block is used to model channel-wise inter-dependencies, essentially acting as a channel-wise attention block. The low representation capacity of binary neural networks can hinder their applicability in general scenarios. Rundo *et al.*

(2019) has shown that an efficient combination of SE blocks in a neural network can improve the overall generalization capability of the networks. It also improves accuracy because of the increased representation capacity of a network. The SE block is a lightweight component and does not add any significant overhead to the overall system when implemented. The overall flow of the quantized convolution layer is shown in Fig. 3.4 where the input and the weights are passed through a quantization function before a convolution layer being applied to it. The SE block works on the original input and is multiplied with the output of the convolution layer to generate the final result.

3.3 Forward and Backward Propagation

During binarization, the vectors lose their Gaussian continuity, which results in a zero value for all the derivatives during backpropagation. Courbariaux *et al.* (2016) and Rastegari *et al.* (2016) (followed by various papers based on the binary neural networks concept) use the Straight-Through Estimator (STE) Bengio *et al.* (2013) to handle the issue of the non-differentiability of the sign function. STE ignores the gradient of the threshold function and treats it as an identity function.

All the layers in the QU-Net model are composed of 1-bit weights except for the last layer. The weights are binarized using the DoReFa-Net Zhou *et al.* (2016) method which uses a sign function as defined in 3.1. Apart from an associated sign function, each weight parameter also has a scalar constant - equal to the mean of all the weight vector values- multiplied with the weight vector. The addition of a mean value vector improves the overall accuracy of the network as it increases the range of values represented by the weight vectors. One thing to note here is that the multiplicative scalar constant does not add any additional overhead as the convolution kernels can still utilize the quantized bit kernels during the convolution operation in both the forward propagation and backward propagation. The full precision vectors are represented by $\mathrm{w_i}$, and the quantized vectors were represented using $\mathrm{w_o}$

- **Forward Pass**

$$w_o = sign(r_i) \cdot mean(abs(w_i)) \tag{3.1}$$

- **Backward Pass**

$$\frac{dl}{dr_i} = \frac{dl}{dr_o}. \tag{3.2}$$

The activation layers followed three different sets of quantization schemes which are based on the DoReFa-Net Zhou *et al.* (2016) method as well. The accuracy was severely degraded when following the same binarization scheme followed by the weights mentioned in the DoReFa-Net paper. To counter this, DoReFa-Net provides an alternative scheme to quantize the activations. The values are initially bound to a range of [0,1], before which the values are scaled down by a factor of 10. The quantization functions as defined in Eq 3.1 and with 3.2 adopted as the backward propagated function. The full precision vectors and the quantized vectors are represented by $\mathrm{r_i}$ and $\mathrm{r_o}$, respectively.

- **Forward Pass**

$$a_o = \frac{1}{2^k - 1} round((2^k - 1)a_i)` \tag{3.3}$$

- **Backward Pass**

$$\frac{dl}{dr_i} = \frac{dl}{dr_o}. \tag{3.4}$$

3.4 Training

QU-Net was trained for a standard 50 epochs on both the COCO and the Cityscapes dataset. The RMSProp optimizer was used with a learning rate of 0.00001, weight decay of 1e-8, and momentum of 0.9. The learning rate used is lower than the one used while training a full precision counterpart since binary networks have their values constrained between the range of [-1,1], and a lower learning rate allows the gradients to accumulate efficiently without exponential increase or decrease. The weights are quantized during each forward pass, but each backpropagation step involves calculating the gradients on the real-valued weight tensors. Gradients generally require a larger bitwidth based on the experiments performed in Zhou *et al.* (2016). Since the model's training did not take much time on full-precision, we did not reduce the bitwidth of the gradients. Images of 320x320 resolution were used to train the model. Although initial experiments were performed using images of size 640x640, it was found that the accuracy decrease was negligible while using images of size 320x320. Using lower resolutions quickly deteriorated the accuracy.

$$\begin{aligned} L_{roi} = &-weight_{ce}[0] * log(exp(x[0])/(\sum_{j} exp(x[j]))) \\ &- weight_{ce}[1] * log(exp(x[1])/(\sum_{j} exp(x[j]))) \\ &sf_{dc}[0] * \frac{\sum p_0 g_0}{\sum p_0 + \sum g_0} + sf_{dc}[1] * \frac{\sum p_1 g_1}{\sum p_1 + \sum g_1} \end{aligned} \tag{3.5}$$

A modified loss function was used to account for the difference in the area occupied by the objects and the background. The overall area occupied by the objects of interest in the scene was much lesser than the background. The low accuracy of the initial experiments when using the original Cross-Entropy + Dice loss can be attributed to the above fact where the validation metrics resulted in the background objects receiving higher importance. Therefore, the loss function was modified to assign more importance to the foreground class. I used a weighted CrossEntropy + scaled Dice loss which helped in accounting for and magnifying the effect of true positives, allowing the capture of the maximum number of regions. However, the loss function resulted in more false positives because the background class was attributed lesser importance.

In the above equation 3.5, the cross-entropy loss is defined by the initial two parts with x representing the class probability and $weight_ce$ representing the weights associated with each class. The weights of each class are equal to the inverse number of pixels containing the class. The dice coefficient is represented by the last two parts of the equation where each class has an associated scale factor (sf_{dc}). p represents the predicted labels in the dice loss, and g represents the actual labels.

3.5 Validation

The final output of the model is a one-hot vector with the values representing the pseudo-probabilities of the two classes - background and foreground. The one-hot vector is compressed to a one-dimensional vector, with each value in a cell representing the row position containing the maximum value. Every position in the one-dimensional vector containing 1 (object positions) represents the region of interest. Segmentation results can result in objects being partially predicted, especially because the boundary regions are difficult to predict. Since QU-Net is being used as a region proposal model, extending the object using a dilation scheme was acceptable as the focus was more on covering every possible region rather than reducing the false positives. The overall model was evaluated based on the three main metrics -

- The number of regions detected with an intersection threshold of 100% - QU-Net is proposed as a region proposal system. The maximum number of regions must be predicted so that the deeper models do not miss these objects. Further, it is of prime importance that the detection covers the maximum area possible such that it does not affect the accuracy of the deeper models. The model is evaluated at the maximum intersection threshold to determine the best-performing model based on correctly predicted and encapsulated regions.

- The amount of area covered by the predicted regions - The proposed model, reduces the overall computation of deeper neural networks, which is achieved by using the predictions by QU-Net to determine the regions of interest. Thus, the least area has to be selected to reduce the computation as much as possible. Even if a particular configuration of the QU-Net model predicts regions with high accuracy but selects a large area, it does not help reduce the overall computation of the deep networks.

- The computational complexity of the model - QU-Net is used in conjunction with deeper models, which warrants the need of having a low complexity model to ensure the overall complexity is not significantly influenced by QU-Net. Although a higher complexity model can reduce the computation of a deeper network with the help of better predictions (correlated to a smaller area), the overall complexity can increase and thus not provide any benefits.

Chapter 4

RESULTS

QU-Net was tested on different scenarios to gauge the overall effectiveness and generalizability of the model. Our experiments showed that QU-Net could be used as an effective region proposal system for different models. Our method ensures that only the relevant regions in the image are processed, effectively reducing the computation of the model by processing a smaller image region.

4.1 Setup

The development of the QU-Net network architecture and the training and testing setup was done using the PyTorch framework. The model was trained on the instance segmentation classes of the Cityscapes dataset and all the classes in the COCO dataset. Each training procedure was run for 50 epochs. The training parameters listed in Section 5.3 were followed for both datasets. QU-Net was trained using images of size 320×320. All the experiments used images of size 640×640 with the image scaled-down and passed through the QU-Net network to obtain the region proposals. The final output of the region proposal model was dilated to ensure the objects were covered and certain surrounding regions were covered sufficiently. This mask was then scaled up and applied to the original image.

4.2 Quantization and DCT-based approach

Initial experiments were performed to understand the overall effectiveness of the model using different computation reduction methodologies, mainly

- Quantization of the U-Net network;

- Using a 2D type II discrete cosine transform on the image to obtain a feature vector in the frequency domain

DCT Input	Initial Layers Quantized (8bit/1bit)	Middle Layers Quantized (1bit/1bit)	Upsampling Layers Quantized (4bit/1bit)	Regions Detected(%)	Area Processed	FLOPs(G)
No	No	No	No	98.36	0.3658	62.68
No	No	Yes	No	98.23	0.3689	55.27
No	No	Yes	Yes	98.23	0.3870	18.36
No	**Yes**	**Yes**	**Yes**	**97.97**	**0.3815**	**5.30**
Yes	Yes	Yes	Yes	99.66	0.9820	0.112

Table 4.1: Results of quantization and application of DCT-II transform

Each combination was trained for 50 iterations. The mAP is calculated on the instance segmentation classes in the Cityscapes dataset.

4.2.1 Quantization based approach

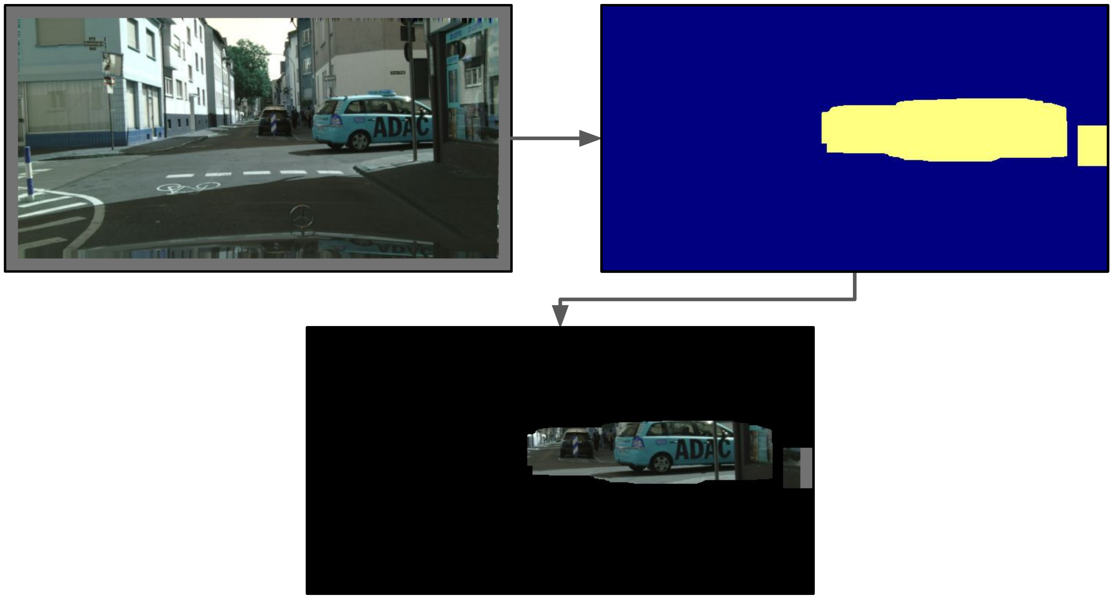


Figure 4.1: The mask predictions by the fully quantized version of U-Net on the top left (yellow is the predicted regions). The bottom row shows the result with the mask applied to the original image.

There have been extensive studies on the effect of quantization on the overall accuracy of the model. Although a lot of works have delved into the effect of quantization on the initial and final layers of the network, very few models have implemented models with the initial layers binarized. This is important as the initial layers can serve as a bottleneck to the rest of the network which is quantized. The main cause of this bottleneck in a convolutional neural network is the multiply-accumulate operations performed as part of the convolution operations. Furthermore, many models require the results to be stored in memory so that deeper layers have access to the previous results (ResNet, for example). This can slow down a network especially when the memory accesses are large. Quantization is an effective method to reduce the overall computation both in terms of memory and computational cost. It reduces the memory footprint by reducing the size of each representation where the 32-bit numbers are represented using 1-bit numbers, thus reducing the model's size by nearly 32 times. The final QU-Net model size occupied a memory space of 2MB, allowing us to fit the model parameters on various resource-constrained devices. Quantizing the model decreases the overall computational cost as shown in Table 4.1. The computation is measured in FLOPs which is the number of floating-point operations performed per second. The overall computation cost of a fully quantized model was reduced by more than ten times compared to the full-precision model. The final results show that the quantized versions of the model show only a marginal decrease in the regions detected with a small increase in the area of the regions proposed. The area captured is 38% of the entire image, equivalent to a similar reduction in the image size processed by the deeper models. The output of the quantized model is shown in 4.1.

4.2.2 DCT based approach

Discrete Cosine Transform (DCT) is a mathematical transformation method that has been widely used in lossy compression techniques such as JPEG. DCT takes

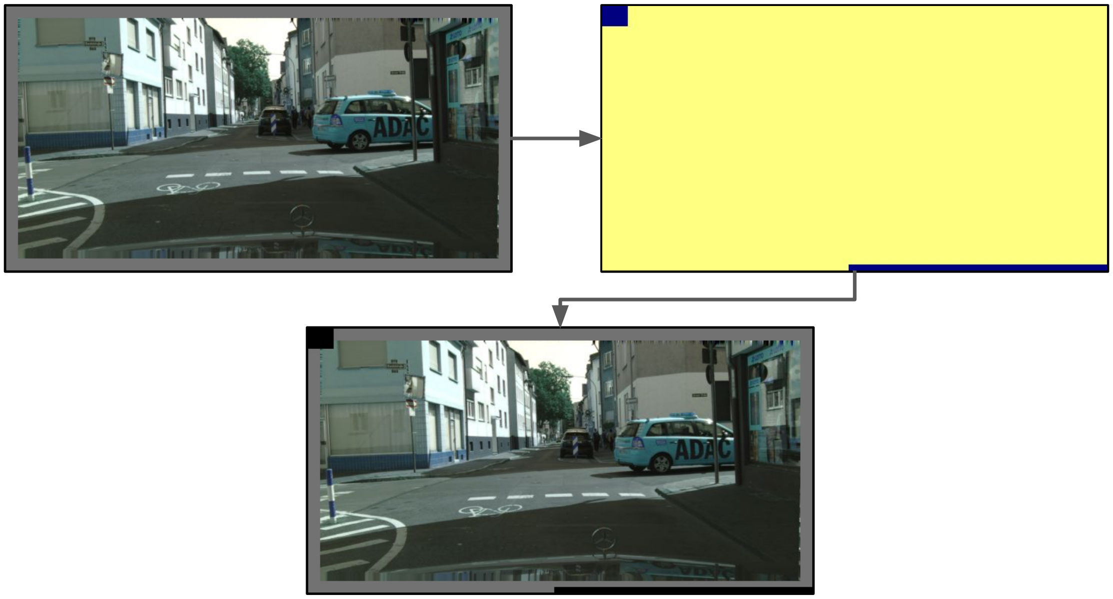


Figure 4.2: The Mask Predictions by the DCT based fully quantized version of U-Net on the top left (yellow is the predicted regions). The bottom row shows the result with the mask applied to the original image.

an image and transforms it from the spatial domain to the frequency domain. The image is generally broken down into a set of 8x8 pixels, and a DCT transform is applied to it. The final result is eight times smaller than the original input. Xu *et al.* (2020) proposed that the DCT method was an effective way to reduce the total computations of the network. The work proposed to compress an image using a DCT transform and use the transformed image to extract the features required for various image recognition tasks. Since the original image size is reduced (by a factor of 8), the overall network complexity also reduces in proportion to it. However, on testing the DCT approach as shown in Table 4.1, it was found that this method performed better than the full-precision method in detecting the regions along with a lower computational cost. However, it selected a large image area for further processing, counter-intuitive to the suggested use-case of QU-Net. This method resulted in more than 98% (as shown in Fig. 4.2) of the image being selected for processing compared to 38% of the area selected for processing by the fully quantized model.

Model	Dataset	Precision	Recall	mAP@0.5	mAP@0.5:0.95	FLOPs(G)
RN101-YOLO	Cityscapes(Person)	68.7	49.5	56.4	32.5	42.29
DC based RN101-YOLO	Cityscapes(Person)	66.2	45.5	51.1	27.2	27.58
RN101-YOLO + QU-Net	Cityscapes(Person)	69.5	45.9	52.9	30.1	21.93
YOLOv5m	Cityscapes	76.6	46.5	53.3	31.1	51.41
YOLOv5m + QU-Net	Cityscapes	78.4	45.1	52	29.7	22.32
YOLOv5m	COCO	69.7	57.5	62.8	43.5	51.41
YOLOv5m + QU-Net	COCO	72.4	53.5	60.0	40.2	42.42
YOLOv5l	COCO	72.8	61.5	66.5	47.2	115.61
YOLOv5l + QU-Net	COCO	72.5	59	63.7	43.8	88.78
YOLOv5x	COCO	74.9	62.7	68.4	49.5	219.02
YOLOv5x + QU-Net	COCO	74.1	60.5	65.5	45.8	163.49

Table 4.2: Results for the object detection task on the Cityscapes and the COCO datasets

4.3 Comparison with Dynamic Convolutional Networks

Dynamic Convolutions Verelst and Tuytelaars (2020) is possibly the closest work. This work generates masks at each ResNet block that predicts the salient regions containing important features and uses the masks generated at each layer to determine the regions where convolutions need to be performed. The proposed method - QU-Net - was compared to the dynamic convolutions approach in two object recognition tasks - object detection and segmentation. Flexible YOLO Yang (2021) and PSPNet Zhao *et al.* (2017) were used as the baseline models. Dynamic convolutions are currently implemented only for ResNet backbones, requiring us to use models which specifically were built using ResNet as their backbone or modify the models to use the ResNet backbones. Dynamic Convolutions defines a computational budget parameter that specifies the number of FLOPs relative to the original model that should be executed. The budget parameter is used to determine the loss at each training step combined with the loss of the network predictions. A budget parameter of 0.25 was chosen, which indicated that 25% of the total FLOPs should be executed. A value of 0.25

was chosen to ensure a fair comparison as a higher value would improve accuracy but would result in a higher number of FLOPs. QU-Net improves upon dynamic convolutions in both object detection and instance segmentation tasks.

4.3.1 Object Detection

The flexible YOLO Yang (2021) model is a model that is built using similar concepts of YOLOv5. YOLOv5 uses a PANet backbone, while flexible YOLO supports a variety of backbones. The ResNet variant, ResNet-101, was used with the dynamic mask supported for the dynamic convolutions approach. The YOLO model with the ResNet backbone is the baseline model against which dynamic convolutions and our method are compared.

The baseline model and the baseline with the dynamic convolutions approach were trained for 200 epochs on the CityPersons detection dataset provided on the official site. CityPersons detection dataset is the object detection dataset for the person class in the Cityscapes dataset. The accuracy was calculated for the best models for each method. Our region proposal system was trained separately on the Cityscapes dataset for 50 epochs. This region proposal method was tested by passing the images through these networks and obtaining a mask used on the original image. Finally, the masked image was passed through the baseline network to determine the final accuracy. Our model achieved a computational reduction of 50% as shown in Table 4.2. RN-101 represents the ResNet-101 backbone and DC represents the Dynamic Convolution approach in the Table 4.2. QU-Net also outperformed the dynamic convolution approach, both in terms of computational complexity as well as accuracy. A mAP accuracy gain of 3% is achieved with an additional computation reduction of 6 GFLOPs lower when compared to the dynamic convolutions.

4.3.2 Segmentation

PSPNet is a segmentation model that achieved state-of-the-art accuracy on various datasets when it was published. It takes into consideration both the global features and the local features to output the final predictions. In addition, the pyramid pooling module is implemented by PSPNet that pools the inputs at various scales after passing them through convolution layers and up-sampling these layers. We use this model as the baseline for the instance segmentation task.

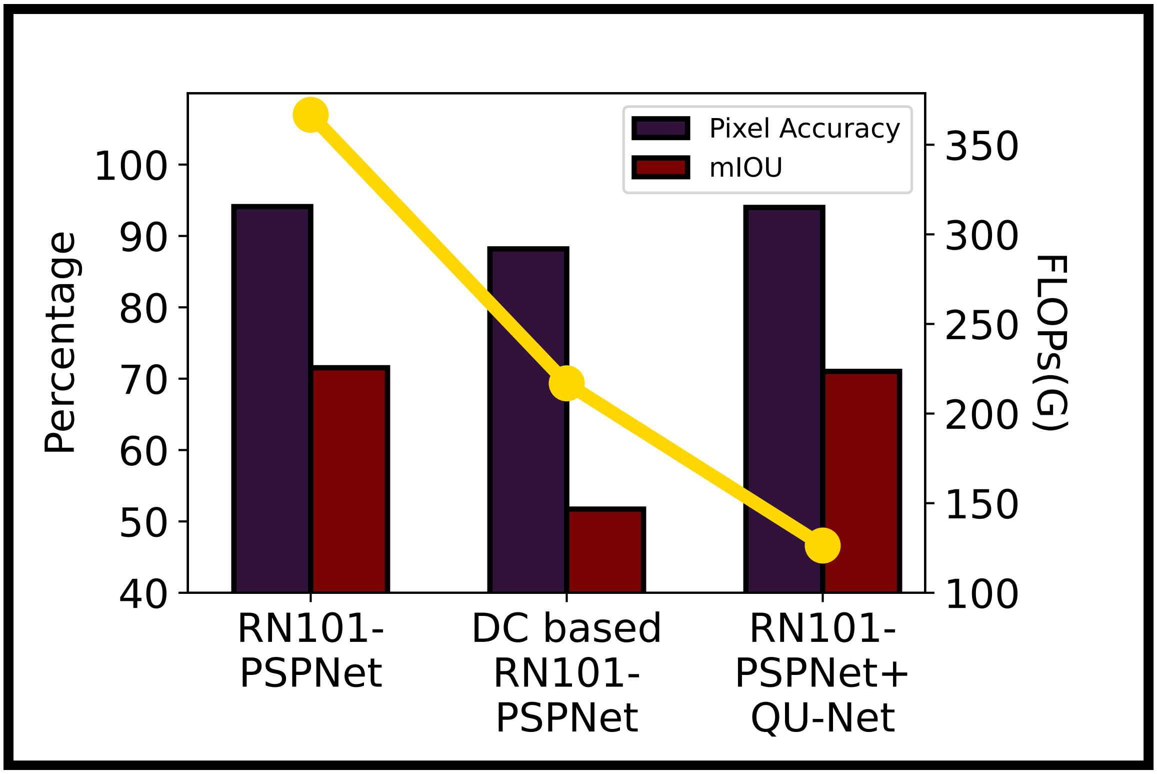


Figure 4.3: Results of the segmentation metrics on the instance segmentation classes in the Cityscapes dataset

The baseline model and the baseline with the dynamic convolutions approach were trained for 200 epochs, similar to the object detection approach. The Cityscapes dataset was used with the instance segmentation classes included. The evaluation metrics were calculated for the best model. The same model used for the object detection tasks was used for the instance segmentation task as well as it does require retraining for different tasks. QU-net provided a better accuracy when compared to the dynamic convolutions model as shown in Fig. 4.3. RN-101 represents the ResNet-101 backbone, and DC represents the Dynamic Convolution approach in Table 4.3. An mAP gain of 20% was achieved along with a reduction in computational cost by 62% compared to the baseline model, 21% more than the dynamic convolutions approach.

4.4 Comparison with state-of-the-art models

Our model is capable of being adapted to different models such as convolutional neural networks and transformers. To test this generalizability, the QU-Net was tested on the YOLOv5 and the Cascade Mask R-CNN models. YOLOv5 is the latest generation of the YOLO family developed by Ultralytics. It is an object detection model which has been pre-trained on the COCO dataset. Cascade Mask R-CNN model is an object detection and instance segmentation model which uses the Swin Transformer as the backbone. The accuracy was computed with and without the addition of the QU-Net region proposal model for both these models.

The comparison for the YOLOv5 network is shown in Table 4.2 while Fig. 4.4) shows the comparison for the Cascade Mask R-CNN model with the Swin transformer backbone. CM R-CNN represents the Cascade Mask R-CNN approach IN THE 4.4 table. The results show a reduction in computation accompanied by a marginal decrease in accuracy. The reduction in the Cityscapes dataset is larger since the objects occupy an area that is four times lesser. For the YOLO model, a computation reduction of 57% and 25% was achieved on the Cityscapes and the COCO datasets, respectively, with an average accuracy reduction of 2%. When implemented on the Cascade Mask R-CNN model, it reduced the number of FLOPs by 114 GFLOPs.

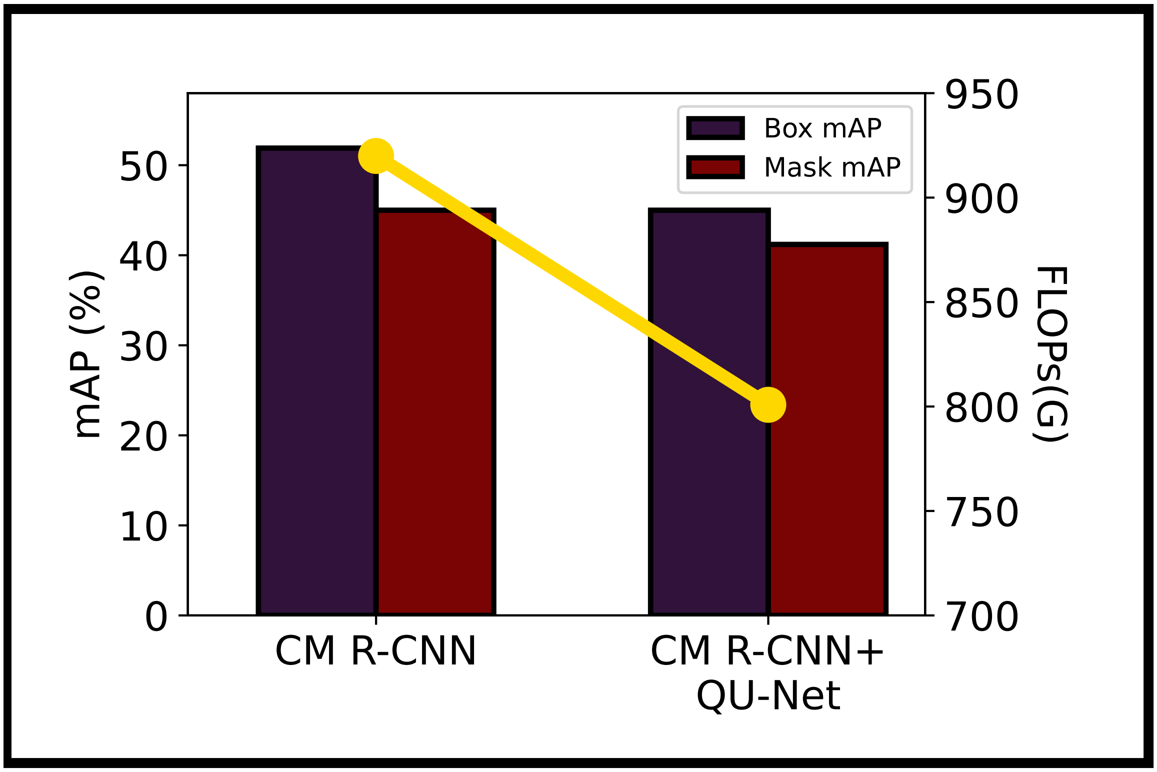


Figure 4.4: Results of the mAP statistics for both object detection and segmentation calculated on the COCO dataset

There was an accuracy reduction of 4.8% for object detection and 3.8% for instance segmentation.

Although QU-Net reduced accuracy, the trade-off factor of reducing the computational cost is more significant with the average computation reduction of 1.5-6 times compared to the average accuracy reduction of 3%. Furthermore, our model is extendable to datasets and can show good performance even when the dataset contains many classes.

4.5 Comparison on variable-scale objects

To understand more about the model's performance with regards to different object scales, experiments were performed on the COCO dataset using the Cascade

Mask R-CNN model with the Swin transformer backbone. These experiments also helped us understand the objects being missed, which allowed us to determine the weaknesses and future improvements of the network. The accuracy (AP) was calculated for each object size based on the following definitions for each scale -

- **Small-scale** - Objects occupying an area less than 32x32 pixels
- **Medium-scale** - Objects occupying an area between 32x32 and 96x96 pixels
- **Large-scale** - Objects occupying an area larger than 96x96 pixels

Model	Dataset	IOU(small)	IOU(medium)	IOU(large)
CM R-CNN	COCO	0.354	0.552	0.673
CM R-CNN + QU-Net	COCO	0.315	0.504	0.621

Table 4.3: Results for the object detection task for different object sizes in the COCO dataset

The initial thought before performing the experiments was that the network misses the small-scale objects, which led to the accuracy degradation. However, the results for the object detection task as shown in Table 4.3 show that that the performance for small-scale objects is close to that of the baseline, with the gap increasing as the object size increase. The difference in accuracy for larger objects being higher can be attributed to the fact that a custom dilation scheme is used for all the regions, irrespective of the size. The post-processing performed by QU-Net can help in the small and medium objects being entirely encapsulated by the image mask, but in the case of large objects, some parts of the object may not be fully covered using a fixed dilation method.

Table 4.4 shows the results for the instance segmentation part where the segmentation results for the large objects lag behind the other two categories. It is also

Model	Dataset	IOU(small)	IOU(medium)	IOU(large)
CM R-CNN	COCO	0.304	0.482	0.608
CM R-CNN + QU-Net	COCO	0.260	0.445	0.557

Table 4.4: Results for the instance segmentation task for different object sizes in the COCO dataset

important to note that it is not sufficient to have a view of the object alone but of the surrounding region to gain further context in specific image recognition tasks. Thus, it can explain the results for the accuracy of the large-scale objects being lower than the other two categories. Future work can also develop a dynamic dilation scheme based on the region's area to ensure the object and the surrounding area are completely encapsulated.

Chapter 5

BIGGER PICTURE - ARGOS

QU-Net is developed as part of a larger framework called ARGOS. ARGOS aims to create an end-to-end framework to run deep neural networks with a high computational cost on low-power devices. In addition, ARGOS aims to reduce the overall computation of the network by proposing the regions that contain objects, allowing deeper models to focus on those regions. This method can help significantly lower the computation cost of deep networks while maintaining accuracy.

ARGOS utilizes two separate systems for proposing the regions containing the objects -

- **QU-Net** - QU-Net is the work detailed in this report that utilizes a binary neural network to output the segmentation labels that can be superimposed on the original image to obtain the masked image

- **Online Knowledge Distillation** - The online knowledge distillation method uses a lighter model to predict the regions, which are iteratively improved from the predictions by the deeper model. The final predictions are a combination of the deeper model and the features obtained from previous frames.

ARGOS proposes two different mechanisms to cater to different scenarios. First, the online knowledge distillation method uses an online learning method to improve the predictions of the lighter network and reduce the computations performed by the deeper network. It uses a motion-based region proposal system which is challenging to extend to scenarios that involve moving cameras or movement in other parts of the scene which may not be of interest, such as the motion of the tree leaves. On

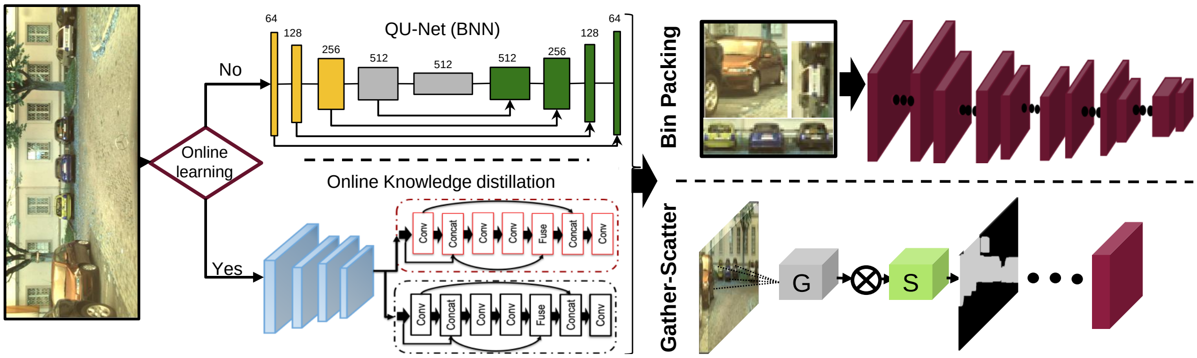


Figure 5.1: Overall ARGOS architecture with the binary region proposal (QU-Net) and the online distillation part

the other hand, QU-Net brings forth the capability to adjust to dynamic situations involving a moving camera or other moving objects that are not important to the final predictions.

As seen in Fig. 5.1, the QU-Net is a part of ARGOS and can be used as a plug-and-play system depending on the requirement. ARGOS seamlessly integrates with different model structures and enables the reduction of the overall computation. Once QU-Net predicts the regions containing the objects, ARGOS can employ two different methods to parse this information. The first one is a bin-packing method. After the light model predicts the regions of interest, the bounding boxes containing the object are cropped from the image and packed (using the *rectpack* algorithm). All the regions are packed into a batch of frames and sent for processing to the deeper network. Although this reduces the total area for processing, it can lead to extraneous areas being selected around the object.

Another method, the gather-scatter method, reduces the computation at the logical blocks level. The different convolution modules are rewritten only to perform computation at the regions of object presence. The inputs are first gathered into a dense matrix filling all the holes where the light model did not predict any object. Once the values are gathered, the convolution operation is performed on the dense matrix. The resulting values are then scattered back to the original locations. A convolution operation is a costly operation because of the large number of MAC operations associated with it. This method can reduce the computational cost of the overall network once the masked input is obtained by implementing custom neural network modules to support the gather-scatter procedure.

Chapter 6

CONCLUSION AND FUTURE WORK

6.1 Conclusion

This research presents a lightweight region detector model used as a black box preprocessing tool for different deep models. The model was trained on both the COCO dataset and the Cityscapes dataset. The model's effectiveness in working with different models such as convolutional neural networks and transformers for different image recognition tasks is effectively displayed. Furthermore, experiments show that QU-Net can be used to reduce the overall computation of the network by reducing the overall area of the image processed.

6.2 Future Work

As part of the future work in extending QU-Net, three avenues can be explored in the future -

- Building an unsupervised model - Currently, all the experiments are performed using a QU-Net model trained on supervised data. Training the model has a dependency on datasets that contain the segmentation labels. The main aim is to use an attention-based mechanism to predict the regions of interest from the feature maps of the image to remove the dependency on labeled data. This improvement can improve the overall application of the model to various datasets.
- Building a 'Smart' Quantization model - QU-Net has been developed by manually experimenting and reasoning the quantization schemes for the different

layers. It can be further explored in the realm of intelligent networks that can predict the best possible quantization scheme for each layer based on the accuracy maintained within a certain threshold.

- Building a Dynamic Dilation Based Model - QU-Net currently employs a standard dilation scheme for regions of all shapes, which causes issues as regions of different sizes can require a different amount of dilation. A future enhancement can be implementing a dynamic dilation scheme based on the model's size, with smaller objects being dilated less than larger ones. The post-processing performed will also ensure the object, and its surrounding is captured, irrespective of its size and reduce the area captured as the regions will fit the objects better.

REFERENCES

Badrinarayanan, V., A. Kendall and R. Cipolla, “Segnet: A deep convolutional encoder-decoder architecture for image segmentation”, CoRR **abs/1511.00561**, URL `http://arxiv.org/abs/1511.00561` (2015).

Banner, R., I. Hubara, E. Hoffer and D. Soudry, “Scalable methods for 8-bit training of neural networks”, arXiv preprint arXiv:1805.11046 (2018).

Bengio, Y., N. Léonard and A. Courville, “Estimating or propagating gradients through stochastic neurons for conditional computation”, (2013).

Chen, L.-C., G. Papandreou, I. Kokkinos, K. Murphy and A. L. Yuille, “Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs”, IEEE transactions on pattern analysis and machine intelligence **40**, 4, 834–848 (2017).

Cordts, M., M. Omran, S. Ramos, T. Rehfeld, M. Enzweiler, R. Benenson, U. Franke, S. Roth and B. Schiele, “The cityscapes dataset for semantic urban scene understanding”, in “Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)”, (2016).

Courbariaux, M., I. Hubara, D. Soudry, R. El-Yaniv and Y. Bengio, “Binarized neural networks: Training deep neural networks with weights and activations constrained to +1 or -1”, (2016).

Figurnov, M., M. D. Collins, Y. Zhu, L. Zhang, J. Huang, D. Vetrov and R. Salakhutdinov, “Spatially adaptive computation time for residual networks”, (2017).

Girshick, R., “Fast r-cnn”, (2015).

Girshick, R., J. Donahue, T. Darrell and J. Malik, “Rich feature hierarchies for accurate object detection and semantic segmentation”, (2014).

Han, Y., G. Huang, S. Song, L. Yang, H. Wang and Y. Wang, “Dynamic neural networks: A survey”, (2021).

Hu, J., 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).

Lim, L. A. and H. Y. Keles, “Learning multi-scale features for foreground segmentation”, Pattern Analysis and Applications **23**, 3, 1369–1380 (2020).

Lim, L. A. and H. Yalim Keles, “Foreground segmentation using convolutional neural networks for multiscale feature encoding”, Pattern Recognition Letters **112**, 256–262, URL `http://dx.doi.org/10.1016/j.patrec.2018.08.002` (2018).

Lin, T.-Y., M. Maire, S. Belongie, L. Bourdev, R. Girshick, J. Hays, P. Perona, D. Ramanan, C. L. Zitnick and P. Dollár, "Microsoft coco: Common objects in context", (2015).

Liu, W., D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu and A. C. Berg, "Ssd: Single shot multibox detector", Lecture Notes in Computer Science p. 21–37, URL `http://dx.doi.org/10.1007/978-3-319-46448-0_2` (2016).

Nguyen, D. V. and J. Choi, "Toward scalable video analytics using compressed-domain features at the edge", Applied Sciences **10**, 18, URL `https://www.mdpi.com/2076-3417/10/18/6391` (2020).

Rastegari, M., V. Ordonez, J. Redmon and A. Farhadi, "Xnor-net: Imagenet classification using binary convolutional neural networks", (2016).

Redmon, J., S. Divvala, R. Girshick and A. Farhadi, "You only look once: Unified, real-time object detection", (2016).

Ren, S., K. He, R. Girshick and J. Sun, "Faster r-cnn: Towards real-time object detection with region proposal networks", (2016).

Ronneberger, O., P. Fischer and T. Brox, "U-net: Convolutional networks for biomedical image segmentation", (2015).

Rundo, L., C. Han, Y. Nagano, J. Zhang, R. Hataya, C. Militello, A. Tangherloni, M. S. Nobile, C. Ferretti, D. Besozzi *et al.*, "Use-net: Incorporating squeeze-and-excitation blocks into u-net for prostate zonal segmentation of multi-institutional mri datasets", Neurocomputing **365**, 31–43 (2019).

Sengar, S. S. and S. Mukhopadhyay, "Moving object detection using statistical background subtraction in wavelet compressed domain", Multimedia Tools and Applications **79**, 9, 5919–5940 (2020).

Su, Y.-C. and K. Grauman, "Leaving some stones unturned: Dynamic feature prioritization for activity detection in streaming video", (2016).

Verelst, T. and T. Tuytelaars, "Dynamic convolutions: Exploiting spatial sparsity for faster inference", (2020).

Wang, H., Y. Xu, B. Ni, L. Zhuang and H. Xu, "Flexible network binarization with layer-wise priority", in "2018 25th IEEE International Conference on Image Processing (ICIP)", pp. 2346–2350 (2018a).

Wang, N., J. Choi, D. Brand, C.-Y. Chen and K. Gopalakrishnan, "Training deep neural networks with 8-bit floating point numbers", (2018b).

Xu, K., M. Qin, F. Sun, Y. Wang, Y.-K. Chen and F. Ren, “Learning in the frequency domain”, (2020).

Yang, L., “flexible-yolov5”, `https://github.com/yl305237731/flexible-yolov5` (2021).

Yeung, S., O. Russakovsky, G. Mori and L. Fei-Fei, “End-to-end learning of action detection from frame glimpses in videos”, (2017).

Zhao, H., J. Shi, X. Qi, X. Wang and J. Jia, “Pyramid scene parsing network”, in “Proceedings of the IEEE conference on computer vision and pattern recognition”, pp. 2881–2890 (2017).

Zhou, S., Y. Wu, Z. Ni, X. Zhou, H. Wen and Y. Zou, “Dorefa-net: Training low bitwidth convolutional neural networks with low bitwidth gradients”, arXiv preprint arXiv:1606.06160 (2016).

APPENDIX A

PERMISSION STATEMENTS FROM CO-AUTHORS

Permission for including co-authored material in this dissertation was obtained from co-authors Mohammad Farhadi, Dr. Yezhou Yang, and Dr. Carole-Jean Wu.