Texas A&M University - Commerce

Department of Computer Science

Automated Blood Cell Identification and Counting

Sridevi Sowmya Grandhi

Supervisor: Derek Harter, Ph.D.

A report submitted in partial fulfilment of the requirements of
Texas A&M University - Commerce for the degree of
Master of Science in *Computer Science*

April 16, 2024

Declaration

I, Sridevi Sowmya Grandhi, of the Department of Computer Science, Texas A&M University - Commerce, confirm that this is my own work and figures, tables, equations, code snippets, artworks, and illustrations in this report are original and have not been taken from any other person's work, except where the works of others have been explicitly acknowledged, quoted, and referenced. I understand that if failing to do so will be considered a case of plagiarism. Plagiarism is a form of academic misconduct and will be penalised accordingly.

I give consent to a copy of my report being shared with future students as an exemplar.

I give consent for my work to be made available more widely to members of TAMUC and public with interest in teaching, learning and research.

Sridevi Sowmya Grandhi
April 16, 2024

Abstract

In our cutting-edge approach to automate the identification and counting of three blood cell types, we employ a sophisticated fusion of deep learning and advanced image processing techniques, particularly focusing on object detection. Traditional complete blood cell counts necessitate laborious manual counting using a haemocytometer, involving intricate laboratory equipment and chemical compounds. This antiquated method is both time-consuming and burdensome. Our innovative solution utilizes Convolutional Neural Networks (CNNs) for intricate feature extraction from microscopic blood sample images. Incorporating state-of-the-art object detection algorithms, such as YOLO (You Only Look Once) or Faster R-CNN (Region-based Convolutional Neural Network), our system precisely identifies and localizes individual blood cells, overcoming the limitations of manual counting. Image processing techniques, including contrast enhancement and morphological operations, are strategically applied to optimize image quality and facilitate accurate object segmentation. This synergistic blend of deep learning and image processing not only expedites the diagnostic process but also significantly improves the accuracy and efficiency of blood cell identification and counting. By automating this intricate task, our approach aims to revolutionize medical diagnostics, providing healthcare professionals with a rapid and reliable tool for comprehensive blood cell analysis.

Keywords: a maximum of five keywords/keyphrase separated by commas

Acknowledgements

An acknowledgements section is optional. You may like to acknowledge the support and help of your supervisor(s), friends, or any other person(s), department(s), institute(s), etc. If you have been provided specific facility from department/school acknowledged so.

Contents

Chapter 1

Introduction

A complete blood cell count (CBC) is vital for assessing health, comprising red blood cells (RBCs), white blood cells (WBCs), and platelets. Manual counting methods are time-consuming and error-prone, requiring automation. Machine learning, particularly deep learning, offers robust solutions across medical applications. Applying deep learning to identify and count blood cells in smear images presents a promising avenue for accurate and efficient analysis, revolutionizing medical diagnostics. Previous models like YOLOv5 and YOLOX have pushed the boundaries of object detection with improved speed and accuracy. YOLOv5 introduced innovations like ConvBN-LeakyReLU and EfficientNet-inspired components. YOLOX further enhanced performance with methods like Cross Stage Partial Network plus CBS. While these models have advanced the field, they may face limitations in handling small objects or complex scenes

1.1 Background

The utilization of image-based methods for disease detection and diagnosis has gained significant attention in recent years. This project focuses on the development of an automated system for the identification and counting of blood cells, leveraging advanced image processing and machine learning techniques.

1.2 Problem statement

The accurate identification and characterization of blood cells, particularly red blood cells (RBCs) and white blood cells (WBCs), pose significant challenges in traditional medical diagnostics. Manual methods for blood cell analysis are often labor-intensive, time-consuming, and prone to human error, leading to variability in results.

1.3 Aims and objectives

Aims: This project aims to develop an automated system for blood cell analysis to improve accuracy, efficiency, and reliability in disease diagnosis.

Objectives: The specific objectives of this project include:

*Exploring existing methodologies for automated blood cell analysis.

*Identifying limitations in current approaches and proposing innovative solutions.

*Designing and implementing a novel solution approach combining image processing and machine learning techniques.

*Evaluating the performance of the developed system and comparing it with existing methods.

1.4 Solution approach

Briefly, the solution approach involves leveraging advanced algorithms and methodologies from image processing and machine learning domains. This includes preprocessing techniques for image enhancement, feature extraction methods, and the implementation of machine learning models for classification and counting of blood cells.

1.5 Summary of contributions and achievements

This study makes several significant contributions to the field of automated blood cell analysis. By addressing key challenges and proposing innovative solutions, we aim to: Improve the accuracy and reliability of blood cell identification and counting. Streamline the analysis process, thereby reducing time and labor requirements. Enhance the diagnostic capabilities of healthcare professionals, leading to improved patient outcomes.

1.6 Organization of the report

The report is organized into several sections for clarity and coherence. It begins with an Introduction, covering background, problem statement, aims, objectives, and solution approach. Following this, the Literature Review discusses pertinent sources and citation practices. Methodology outlines the research methodology adopted. Results present the findings obtained from the study. Discussion and Analysis critically analyze the results, highlighting their significance and limitations. Conclusions summarize the key findings and suggest avenues for future research. Finally, Appendices include supplementary materials such as data tables or additional information for interested readers. This structure aims to guide readers through the report's content efficiently and comprehensively.

Chapter 2

Literature Review

In recent times, there have been big improvements in how we analyze blood cell images. These advancements have made counting cells easier and more accurate. For instance, one study created a smart way to count red blood cells using special image tricks. Another study found a better way to spot objects in images, which helps count cells more precisely. Some researchers also figured out how to find unusual cells by looking at their shape and color in microscope pictures. They even made a cool new method to find round cells in images, which is super helpful for counting red blood cells. Plus, there's a clever computer model that's learning to count blood cells all on its own. These new ideas are making blood cell analysis faster and more reliable.

2.1 Example of in-text citation of references in LaTeX

Alomari et al. (n.d.)

2.2 Example of "risk" of unintentional plagiarism

Navigating the landscape of academic writing requires a keen awareness of the potential pitfalls, and one significant challenge is the risk of unintentional plagiarism. This occurs when writers inadvertently mismanage the incorporation of external sources, ideas, or materials into their work, leading to improper paraphrasing, summarizing, quoting, or citing. The nuances of proper attribution can be intricate, and unintentional plagiarism often stems from a lack of awareness or failure to adhere to citation rules.

Example of Unintentional Plagiarism – Citing Wrongly:

A common illustration of unintentional plagiarism is the improper citation of sources. Imagine a scenario where a writer, in the process of compiling research, encounters a compelling idea from a scholarly article. While attempting to integrate this idea into their work, they inadvertently misattribute it to another source or overlook the necessity of proper citation. This misstep results in unintentional plagiarism, as the writer fails to give due credit to the original author. Whether due to oversight, unfamiliarity with citation guidelines, or misinterpretation of the source, such instances underscore the importance of meticulous citation practices to avoid unintentional plagiarism and uphold the principles of academic integrity.

2.3 Critique of the review

In recent advancements in blood cell image analysis, several studies have significantly contributed to automating and enhancing the accuracy of blood cell counting processes. One notable study Alomari et al. (n.d.), focuses on applying image processing techniques to extract blood cell images from microscopes, particularly automating the red blood cell counting process. Through the utilization of digital image processing, the study employs an edge detection algorithm to identify and count red blood cells, demonstrating a pivotal advancement in the field. Another noteworthy approach Maitra et al. (2012), involves the application of the Hough transform, a well-established feature extraction technique. Initially developed for line detection, this method has been extended for detecting low-parametric objects, such as circles. While offering a cost-effective and efficient approach, the study suggests the need for modifications to ensure accurate counting, stressing the necessity for further investigations into complete blood cell counts.

In the realm of nuclei extraction, Poomcokrak and Neatpisarnvanit (2008) employ clustering of microscopic images and the curvelet transform, proving effective in detecting detailed information and enabling discrimination between atypical and blast cases. The study introduces a novel feature, the color saturation gradient, contributing to the classification of lymphoblast cells and atypical lymphoma cells. Another significant contribution comes from Putzu and Di Rubert (2013), where it proposes a method for leukocyte segmentation and identification. This approach integrates pre-processing methods to simplify and enhance segmentation, emphasizing the multi-stage process, including shape control and nucleus-cytoplasm selection, contributing to more robust leukocyte identification.Utilizing thresholding and morphological operations, the circularity feature Sarrafzadeh et al. (2015) of blood cells is employed in an iterative structured circle detection algorithm. This introduces a new technique for binary image separation and demonstrates promising results.

Further innovations include the introduction of the Circlet Transform Soltanzadeh et al. (2012), offering a novel method for segmenting circular objects, with a specific focus on red blood cells. Utilizing the Circular Hough Transform, the method showcases potential in RBC segmentation.

2.4 Summary

In recent strides towards automating blood cell image analysis, a series of noteworthy studies have significantly advanced the field. These studies encompass diverse methodologies, such as image processing, Hough transform, clustering, and deep learning. One study pioneers the use of image processing, specifically an edge detection algorithm, to automate red blood cell counting, marking a pivotal advancement. Another approach employs the Hough transform for feature extraction, offering a cost-effective method for detecting low-parametric objects, though suggesting the need for modifications for accurate counting. Further contributions involve nuclei extraction using clustering and the curvelet transform, introducing a novel feature for discriminating atypical cases. Another study proposes a multi-stage method for leukocyte segmentation, emphasizing shape control and nucleus-cytoplasm selection. Innovations also include the introduction of the Circlet Transform for segmenting circular objects, with a focus on red blood cells, and the application of a deep learning model GitHub, Inc. (2020) trained on the Blood Cell Count Dataset, showcasing promising results in automating blood cell counting. Collectively, these studies underscore the

diverse and evolving landscape of techniques contributing to the automation and accuracy of blood cell image analysis.

Chapter 3

Methodology

3.1 Algorithms descriptions

Region-based Convolutional Neural Networks (R-CNN): R-CNN is a seminal method for object detection that works by generating region proposals followed by CNN-based feature extraction for each proposal. Initially proposed by Girshick et al. [8], R-CNN has laid the foundation for subsequent advancements in object detection.

Fast R-CNN: Fast R-CNN [9] improves upon the R-CNN framework by integrating the region proposal mechanism into the network architecture, enabling end-to-end training and faster inference.

Faster R-CNN: Building upon Fast R-CNN, Faster R-CNN [10] introduces a Region Proposal Network (RPN) to efficiently generate region proposals, making the entire detection pipeline faster and more accurate.

EfficientDet: EfficientDet [11] is a scalable and efficient object detection model that achieves state-of-the-art performance by optimizing model architecture and scaling strategies. You Only Look Once (YOLO) series (YOLOv1-v8):

You Only Look Once (YOLO) series (YOLOv1-v8):

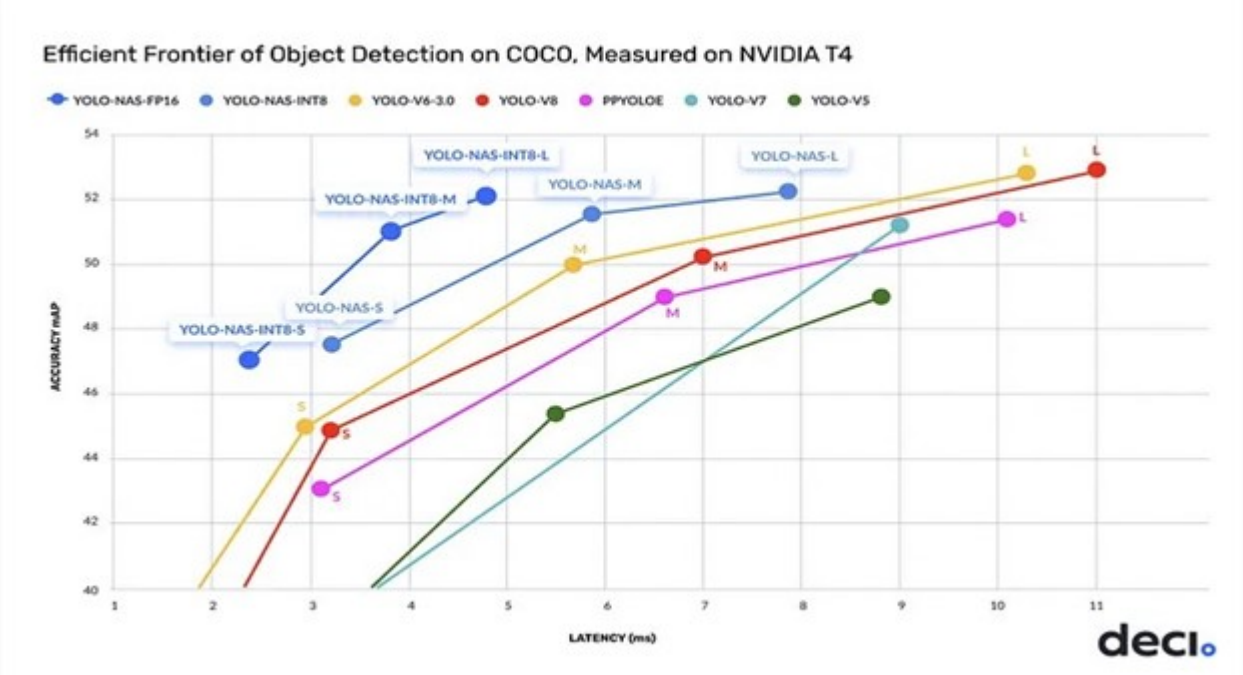

Figure 3.1: Figure 3.1 Efficient frontier comparison of YOLO models

The YOLO series represents a significant advancement in object detection methodology, particularly known for its real-time inference capabilities. YOLOv1 [12] introduced the concept of dividing the input image into a grid and predicting bounding boxes and class probabilities directly from the grid cells. This approach enables YOLO to achieve remarkable speed while maintaining competitive accuracy.

Subsequent iterations of YOLO, including YOLOv2, YOLOv3, YOLOv4, and YOLOv5, have introduced various improvements in terms of speed, accuracy, and model architecture. YOLOv2 [13] introduced the concept of anchor boxes for better localization, while YOLOv3 [14] improved upon its predecessor with feature pyramid networks and multi-scale predictions.

YOLOv4 [15] further enhanced the model's performance by incorporating techniques like CSPDarknet53, PANet, and SPP. It achieved state-of-the-art results in terms of both speed and accuracy. YOLOv5 introduced a streamlined architecture and advanced data augmentation techniques, resulting in improved performance and efficiency.

The YOLO series has continued to evolve with the introduction of YOLOv6 and YOLOv7, each bringing novel innovations to the field of object detection. These iterations have focused on optimizing model architectures, training strategies, and inference speed to address various challenges in real-world applications.

YOLO NAS (Neural Architecture Search): YOLO NAS [16] represents a departure from traditional hand-designed architectures by leveraging Neural Architecture Search (NAS) techniques to automatically discover optimal network architectures. By searching through a predefined search space of architectural components, YOLO NAS can tailor the model architecture to specific datasets and tasks, leading to improved performance and efficiency.

3.2 Code

:

```
import shutil
import os, sys, random
import xml.etree.ElementTree as ET
from glob import glob
import pandas as pd
from shutil import copyfile
import pandas as pd
from sklearn import preprocessing, model_selection
import matplotlib.pyplot as plt
%matplotlib inline
from matplotlib import patches
import numpy as np
import os

labels = sorted(glob('/content/BCCD_Dataset/BCCD/Annotations/*.xml'))
# print(len(labels))
df = []
total = 0
for file in labels:
  prev_filename = file.split('/')[-1].split('.')[0] + '.jpg'
  filename = str(total) + '.jpg'
  row = []
  parsedXML = ET.parse(file)
  for node in parsedXML.getroot().iter('object'):
    blood_cells = node.find('name').text
    xmin = int(node.find('bndbox/xmin').text)
    xmax = int(node.find('bndbox/xmax').text)
    ymin = int(node.find('bndbox/ymin').text)
    ymax = int(node.find('bndbox/ymax').text)

    row = [prev_filename, filename, blood_cells, xmin, xmax, ymin, ymax]
    df.append(row)
  total += 1

data = pd.DataFrame(df, columns=['prev_filename', 'filename', 'cell_type', '
    xmin', 'xmax', 'ymin', 'ymax'])

data[['prev_filename','filename', 'cell_type', 'xmin', 'xmax', 'ymin', 'ymax'
    ]].to_csv('/content/blood_cell_detection.csv', index=False)

img_width = 640
img_height = 480

def width(df):
  return int(df.xmax - df.xmin)
def height(df):
  return int(df.ymax - df.ymin)
def x_center(df):
  return int(df.xmin + (df.width/2))
def y_center(df):
  return int(df.ymin + (df.height/2))
def w_norm(df):
```

```
  return df/img_width
def h_norm(df):
  return df/img_height

df = pd.read_csv('/content/blood_cell_detection.csv')
print(len(df))
le = preprocessing.LabelEncoder()
le.fit(df['cell_type'])
print(le.classes_)
labels = le.transform(df['cell_type'])
df['labels'] = labels

df['width'] = df.apply(width, axis=1)
df['height'] = df.apply(height, axis=1)

df['x_center'] = df.apply(x_center, axis=1)
df['y_center'] = df.apply(y_center, axis=1)

df['x_center_norm'] = df['x_center'].apply(w_norm)
df['width_norm'] = df['width'].apply(w_norm)

df['y_center_norm'] = df['y_center'].apply(h_norm)
df['height_norm'] = df['height'].apply(h_norm)

df.head(30)

df.describe()

df.tail(10)

import cv2
def show_mask(image_name):
  """A function to display image and bounding box"""
  fig = plt.figure(figsize=(6.5, 6.5))
  ax = fig.add_axes([0,0,1,1])
  image = plt.imread('/content/BCCD_Dataset/BCCD/JPEGImages/BloodImage_0000'
    + image_name)
  plt.imshow(image)

  # ax.imshow(image)

  for _, row in df[df.filename == image_name].iterrows():
        xmin = row.xmin
        xmax = row.xmax
        ymin = row.ymin
        ymax = row.ymax

        width = xmax - xmin
        height = ymax - ymin

        if row.cell_type == 'RBC':
            edgecolor = 'r'
            ax.annotate('RBC', xy=(xmax-40, ymin+20))
        elif row.cell_type == 'WBC':
            edgecolor = 'b'
            ax.annotate('WBC', xy=(xmax-40, ymin+20))
```

```
        elif row.cell_type == 'Platelets':
            edgecolor = 'g'
            ax.annotate('Platelets', xy=(xmax-40, ymin+20))

        rect = patches.Rectangle((xmin, ymin), width, height, edgecolor=
    edgecolor, facecolor='none')
        ax.add_patch(rect)
  plt.title("Blood Cell Types : RBC , WBC , Platelets")
  plt.text(0.5, -0.1, f"Image Width", horizontalalignment='center',
    verticalalignment='center', transform=plt.gca().transAxes)
  plt.text(-0.1, 0.5, f"Image Height", horizontalalignment='center',
    verticalalignment='center', rotation=90, transform=plt.gca().transAxes)
 plt.show()

show_mask('1.jpg')
show_mask('2.jpg')
show_mask('3.jpg')

Dataset Distribution

from sklearn.model_selection import train_test_split
df_train_temp, df_valid = train_test_split(df, test_size=0.10, random_state
    =42, shuffle=True, stratify=df['cell_type'])
df_train, df_test = train_test_split(df_train_temp, test_size=0.075,
    random_state=42, shuffle=True, stratify=df_train_temp['cell_type'])

Data Visualization

"""Plotting of total instances of RBC ,WBC ,Platelets in Train,Validation an
    Test Dataset"""
plt.subplot(1, 3, 2)
plt.bar(valid_class_counts.index, valid_class_counts.values)
plt.title('Validation Set Class Distribution')
plt.xlabel('Classes')
plt.ylabel('Instance Count')

plt.subplot(1, 3, 3)
plt.bar(test_class_counts.index, test_class_counts.values)
plt.title('Test Set Class Distribution')
plt.xlabel('Classes')
plt.ylabel('Instance Count')

plt.tight_layout()
plt.show()

""Plotting total count of images in train, validation, and test sets"""
plt.figure(figsize=(10, 5))

labels = ['Train', 'Validation', 'Test']
values = [train_total_images, valid_total_images, test_total_images]

plt.bar(labels, values, color=['orange', 'blue', 'red'])
plt.title('Total Count of Images in Train, Validation, and Test Sets')
plt.xlabel('Datasets')
plt.ylabel('Image Count')
```

```
155 plt.show()
```

Listing 3.1: Code snippet in LaTeX and this is a Python code

3.3 Implementation

YOLO NAS Model Selection: YOLO NAS model architecture was used for object detection experiments. Consider YOLO NAS-S, YOLO NAS-M, and YOLO NAS-L variants. **Training Configuration**: Configured the training parameters including the number of epochs (50-100), batch size (8,16,32), optimizer (e.g., Adam), learning rate schedule (e.g., cosine annealing with warm-up), loss function (e.g., YOLO loss), and any additional metrics for evaluation (e.g., mAP, precision, recall).

You Only Look Once (YOLO) series (YOLOv1-v8):

Dataset Preparation: The dataset was prepared for training, validation , and testing . This includes data preprocessing steps such as annotation extraction, data augmentation, and splitting the dataset into training (80**Inference**: Set up the inference pipeline for performing object detection on new images using the trained YOLO NAS models. **Evaluation**: Implement the evaluation pipeline to calculate performance metrics such as mAP@0.50:0.95, precision@0.50:0.95, and recall@0.50:0.95 on the test dataset.

3.4 Experiments Design

: **Model Evaluation**: Evaluate YOLO NAS models (YOLO NAS-S, YOLO NAS-M, YOLO NAS-L) on the selected metrics (mAP@0.50:0.95, precision, recall) for different configurations. **Hyperparameter Tuning**: Experiment with different hyperparameters such as input image sizes, number of epochs, and quantization levels (8-bit, 16-bit, 32-bit). **Input Image Sizes**: Experiment with different input image sizes to observe their impact on model performance and inference speed. **Quantization Levels**: Experiment with different quantization levels (8-bit, 16-bit, 32-bit) to analyze the trade-off between model accuracy and computational efficiency. **Epochs**: Train the models for different numbers of epochs (e.g., 50, 75, 100) to observe the convergence behavior and model performance over time. **Comparison with YOLO-S, YOLO-M, YOLO-L**: Compare the performance of YOLO NAS models with standard YOLO variants (YOLO-S, YOLO-M, YOLO-L) on the specified metrics.

3.5 Experimental Procedure

: Initialize the selected YOLO NAS model architecture. Train the model using the training dataset with the specified configurations (epochs, batch size, etc.). Validate the trained model on the validation dataset to monitor performance metrics. Fine-tune the model if necessary based on validation results.Evaluate the final trained model on the test dataset to report the performance metrics (mAP@0.50:0.95, precision, recall). Repeat the experiments for different configurations and variations.

3.6 Results Analysis

: Analyze the experimental results to identify the optimal YOLO NAS model configuration based on the specified metrics.

Compare the performance of YOLO NAS models with standard YOLO variants and observe any differences in accuracy, speed, and efficiency.

Discuss the impact of hyperparameters (input image sizes, epochs, quantization levels) on model performance and computational efficiency.

Interpret the results to draw conclusions and insights regarding the effectiveness of YOLO NAS for object detection tasks.

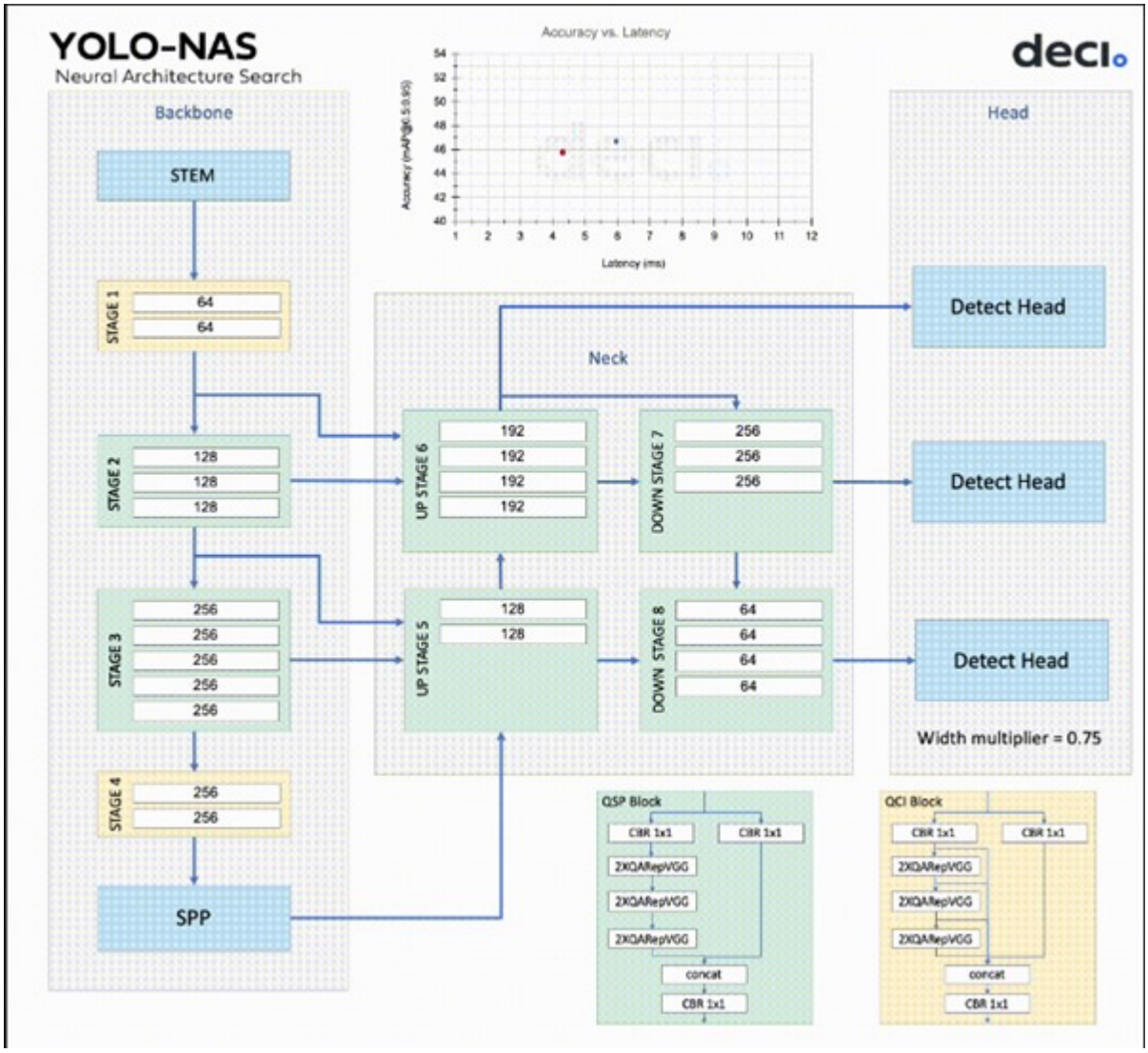

Figure 3.2: Figure 3.2 : YOLO-NAS Neural Architecture Search

Chapter 4

Results

The results chapter tells a reader about your findings based on the methodology you have used to solve the investigated problem. For example:

- If your project aims to develop a software/web application, the results may be the developed software/system/performance of the system, etc., obtained using a relevant methodological approach in software engineering.
- If your project aims to implement an algorithm for its analysis, the results may be the performance of the algorithm obtained using a relevant experiment design.
- If your project aims to solve some problems/research questions over a collected dataset, the results may be the findings obtained using the applied tools/algorithms/etc.

Arrange your results and findings in a logical sequence.

4.1 A section

...

4.2 Example of a Table in LaTeX

Table 4.1 is an example of a table created using the package LaTeX"booktabs." do check the link: wikibooks.org/wiki/LaTeX/Tables for more details. A table should be clean and readable. Unnecessary horizontal lines and vertical lines in tables make them unreadable and messy. The example in Table 4.1 uses a minimum number of liens (only necessary ones). Make sure that the top rule and bottom rule (top and bottom horizontal lines) of a table are present.

Table 4.1: Example of a table in LaTeX

Bike		
Type	Color	Price (£)
Electric	black	700
Hybrid	blue	500
Road	blue	300
Mountain	red	300
Folding	black	500

4.3 Example of captions style

- The **caption of a Figure (artwork) goes below** the artwork (Figure/Graphics/illustration). See example artwork in Figure **??**.
- The **caption of a Table goes above** the table. See the example in Table 4.1.
- The **caption of an Algorithm goes above** the algorithm. See the example in Algorithm **??**.
- The **caption of a Listing goes below** the Listing (Code snippet). See example listing in Listing 3.1.

4.4 Summary

Write a summary of this chapter.

Chapter 5

Discussion and Analysis

Depending on the type of project you are doing, this chapter can be merged with "Results" Chapter as " Results and Discussion" as suggested by your supervisor.

In the case of software development and the standalone applications, describe the significance of the obtained results/performance of the system.

5.1 A section

Discussion and analysis chapter evaluates and analyses the results. It interprets the obtained results.

5.2 Significance of the findings

In this chapter, you should also try to discuss the significance of the results and key findings, in order to enhance the reader's understanding of the investigated problem

5.3 Limitations

Discuss the key limitations and potential implications or improvements of the findings.

5.4 Summary

Write a summary of this chapter.

Chapter 6

Conclusions and Future Work

6.1 Conclusions

Typically a conclusions chapter first summarizes the investigated problem and its aims and objectives. It summaries the critical/significant/major findings/results about the aims and objectives that have been obtained by applying the key methods/implementations/experiment set-ups. A conclusions chapter draws a picture/outline of your project's central and the most signification contributions and achievements.

A good conclusions summary could be approximately 300–500 words long, but this is just a recommendation.

A conclusions chapter followed by an abstract is the last things you write in your project report.

6.2 Future work

This section should refer to Chapter 4 where the author has reflected their criticality about their own solution. The future work is then sensibly proposed in this section.

Guidance on writing future work: While working on a project, you gain experience and learn the potential of your project and its future works. Discuss the future work of the project in technical terms. This has to be based on what has not been yet achieved in comparison to what you had initially planned and what you have learned from the project. Describe to a reader what future work(s) can be started from the things you have completed. This includes identifying what has not been achieved and what could be achieved.

A good future work summary could be approximately 300–500 words long, but this is just a recommendation.

Chapter 7

Reflection

Write a short paragraph on the substantial learning experience. This can include your decision-making approach in problem-solving.

Some hints: You obviously learned how to use different programming languages, write reports in LaTeXand use other technical tools. In this section, we are more interested in what you thought about the experience. Take some time to think and reflect on your individual project as an experience, rather than just a list of technical skills and knowledge. You may describe things you have learned from the research approach and strategy, the process of identifying and solving a problem, the process research inquiry, and the understanding of the impact of the project on your learning experience and future work.

Also think in terms of:

- what knowledge and skills you have developed
- what challenges you faced, but was not able to overcome
- what you could do this project differently if the same or similar problem would come
- rationalize the divisions from your initial planed aims and objectives.

A good reflective summary could be approximately 300–500 words long, but this is just a recommendation.

Note: The next chapter is "**References**," which will be automatically generated if you are using BibTeX referencing method. This template uses BibTeX referencing. Also, note that there is difference between "References" and "Bibliography." The list of "References" strictly only contain the list of articles, paper, and content you have cited (i.e., refereed) in the report. Whereas Bibliography is a list that contains the list of articles, paper, and content you have cited in the report plus the list of articles, paper, and content you have read in order to gain knowledge from. We recommend to use only the list of "References."

References

Alomari, Y. M., Abdullah, S. N. H. S., Zaharatul Azma, R. and Omar, K. (n.d.), 'Automatic detection and quantification of wbcs and rbcs using iterative structured circle detection algorithm'. Published online. Cited by other articles.

GitHub, Inc. (2020), 'Blood cell count dataset', `https://github.com/Shenggan/BCCD_Dataset`.

Maitra, M., Gupta, R. K. and Mukherjee, M. (2012), 'Detection and counting of red blood cells in blood cell images using hough transform', *International Journal of Computer Applications* **53**(16).

Poomcokrak, J. and Neatpisarnvanit, C. (2008), 'Red blood cells extraction and counting', *Department of Biomedical Engineering, Mahidol University, Thailand. The 3rd International Symposium on Biomedical Engineering (ISBME 2008)* .

Putzu, L. and Di Rubert, C. (2013), 'White blood cells identification and counting from microscopic blood image', *World Academy of Science, Engineering and Technology* **73**.

Sarrafzadeh, O., Dehnavi, A. M., Rabbani, H., Ghane, N. and Talebi, A. (2015), 'Circlet based framework for red blood cells segmentation and counting'.

Soltanzadeh, R., Rabbani, H. and Talebi, A. (2012), 'Extraction of nucleolus candidate zone in white blood cells of peripheral blood smear images using curvelet transform', *Computational and mathematical methods in medicine* **2012**.

Appendix A

An Appendix Chapter (Optional)

Some lengthy tables, codes, raw data, length proofs, etc. which are **very important but not essential part** of the project report goes into an Appendix. An appendix is something a reader would consult if he/she needs extra information and a more comprehensive understating of the report. Also, note that you should use one appendix for one idea.

An appendix is optional. If you feel you do not need to include an appendix in your report, avoid including it. Sometime including irrelevant and unnecessary materials in the Appendices may unreasonably increase the total number of pages in your report and distract the reader.

Appendix B

An Appendix Chapter (Optional)

...