Development of virtual immunofluorescence images from hematoxylin and eosin-stained images for cancer diagnosis

Azam, Abu Bakr

2021

Azam, A. B. (2021). Development of virtual immunofluorescence images from hematoxylin and eosin-stained images for cancer diagnosis. Master's thesis, Nanyang Technological University, Singapore. <https://hdl.handle.net/10356/152688>

<https://hdl.handle.net/10356/152688>

<https://doi.org/10.32657/10356/152688>

This work is licensed under a Creative Commons Attribution-NonCommercial 4.0 International License (CC BY-NC 4.0).

Downloaded on 26 Apr 2023 23:29:51 SGT

**NANYANG
TECHNOLOGICAL
UNIVERSITY**

SINGAPORE

**DEVELOPMENT OF VIRTUAL IMMUNOFLUORESCENCE IMAGES
FROM HEMATOXYLIN AND EOSIN-STAINED IMAGES FOR
CANCER DIAGNOSIS**

**AZAM, ABU BAKR
SCHOOL OF MECHANICAL AND AEROSPACE ENGINEERING
2021**

**DEVELOPMENT OF VIRTUAL IMMUNOFLUORESCENCE IMAGES
FROM HEMATOXYLIN AND EOSIN-STAINED IMAGES FOR
CANCER DIAGNOSIS**

AZAM, ABU BAKR

**School of Mechanical and Aerospace Engineering
College of Engineering
Nanyang Technological University**

**A thesis submitted to the Nanyang Technological University
in partial fulfilment of the requirements for the degree of
Master of Engineering**

2021

Statement of Originality

I hereby certify that the work embodied in this thesis is the result of original research, is free of plagiarised materials, and has not been submitted for a higher degree to any other University or Institution.

17th Aug 2021

Date

A handwritten signature in black ink, which appears to read 'Azam Abu Bakr', is enclosed within a hand-drawn oval. The signature is written in a cursive style.

Azam Abu Bakr

Supervisor Declaration Statement

I have reviewed the content and presentation style of this thesis and declare it is free of plagiarism and of sufficient grammatical clarity to be examined. To the best of my knowledge, the research and writing are those of the candidate except as acknowledged in the Author Attribution Statement. I confirm that the investigations were conducted in accord with the ethics policies and integrity standards of Nanyang Technological University and that the research data are presented honestly and without prejudice.

17th Aug 2021

Date

NTU NTU NTU NTU NTU NTU NTU NTU
NTU NTU NTU NTU NTU NTU NTU NTU
NTU NTU NTU NTU NTU NTU NTU NTU
NTU NTU NTU NTU NTU NTU NTU NTU

Assoc Prof Cai Yiyu

Authorship Attribution Statement

Chapters 4 and 5 were used as part of a topical presentation on the topic of AI for Medical Imaging Computing and Computer-assisted Intervention at NTU - NVIDIA NVAITC Joint Symposium on Artificial Intelligence on February 16, 2021.

1. The part of the presentation with this study in focus was prepared by me.
2. The contents were presented by Prof Cai.

Parts of chapters 1, 2 and 3 and the entirety of chapters 4 and 5 would be used to present at the Society of Immuno-Therapy of Cancer's 36th Annual Meeting & Pre-Conference Programs, November 10-14, 2021, Washington, DC. and subsequently publish in its peer-reviewed journal, bearing the title:

Azam A.B., Chang Y.Q., Matthew L., Lim J., Lau M., Tan B., Goh D., Huang L., Yeong J., Cai Y. (2021), Using deep learning approaches with mIF images to enhance T cell identification for tumour -Automation of infiltrating lymphocytes (TILs) scoring on H&E images

1. I would be preparing the initial manuscript drafts. Prof Cai and Dr Yeong will be reviewing the article.
2. I will be presenting the article in the conference.
3. I was responsible for the design and development of the study, as well as the conduction of experiments.

17th Aug 2021

Date

Azam Abu Bakr

Abstract

Haematoxylin and Eosin (H&E) staining is common and viewing these stained images under brightfield microscopes provide basic information of the tumours and other nuclei. In contrast, Immunohistochemical (IHC) images are crucial for cancer diagnosis as it could reveal more information about tumours and its response to treatment. Multiplex Immunofluorescence (mIF), a part of IHC provides a more detailed understanding of the tumour using darkfield microscopy and florescent cameras as opposed to RGB cameras with special monoclonal antibody-based stains. This helps pathologists focus on multiple “biomarkers” or indicators of certain biological processes like immune response. If the same biopsy specimen is used for inspection, the related features obtained from H&E staining and multiplex IF can be utilized to create a Computer Aided Diagnosis (CAD) system including Convolutional Neural Networks which are popularly used in object detection and image segmentation tasks.

The study is divided into two parts, automated optical flow-based image registration and CD3 (biomarker for T cells) region prediction using a special type of a convolutional neural network called as generative adversarial networks (GANs). Concepts of optical flow, k-means clustering, and Otsu thresholding are combined to create a faster and robust intensity-based image registration pipeline, in which the DAPI (4',6-diamidino-2-phenylindole) channel of the mIF image is co-registered with the corresponding H&E image, following which the other channels in mIF image are transformed to match the registration. Finally, the CD3 channel image is superimposed with the matching H&E image to create the reference image needed for deep learning. Two variations of GAN, the Pix2Pix GAN and cycleGAN models are modified to work with the registered image dataset to predict CD3 regions.

As mIF images are available only by using expensive and complex machines, inexpensive and easy to obtain H&E images can now be used in conjunction with GAN models to obtain similar data, which could significantly reduce the costs of cancer treatment since this method not only helps in getting multi-modal image data based on only one type of image, but it also helps in making cancer immunotherapy, a form of cancer treatment dependent on these images mainstream.

Acknowledgements

I would like to thank Prof. Yiyu Cai, my main advisor who provided me with valuable support while I was learning the fundamentals necessary to do the project and use the devices which I have never used before with some help from his team, who made me feel like one of their own. Next, I would like to thank my co-advisor Dr. Joe Yeong, who helped me to understand medical concepts with the help of his team and provided me with the dataset necessary to experiment on. Under their skilful guidance, I was able to prepare unique solutions for pathological image registration and generation of virtual pathological images using advanced machine learning concepts.

Lastly, I would like to thank my parents, my brother and other well-wishers who have supported me in all the ways possible, especially to keep my spirits high during the period of COVID-19.

Table of Contents

Statement of Originality	iii
Supervisor Declaration Statement.....	iv
Authorship Attribution Statement	v
Abstract.....	vi
Acknowledgements	vii
List of Tables.....	x
List of Figures	xi
1. Introduction.....	13
1.1 Definition of Cancer and Related Statistics	13
1.2 Tumour Microenvironments.....	15
1.3 Clusters of Differentiation (CD).....	18
1.4 Cancer Diagnosis	19
1.5 Histopathology and Cancer	20
1.6 Cancer Treatment Expenses	25
1.7 Objective and Scope	26
1.8 Thesis Organization	26
2. Literature Survey.....	28
2.1 Image Registration	28
2.2 Medical Imaging Using Deep Learning.....	30
3. AI Related Fundamentals	34
3.1 Image Registration	34
3.2 Gaussian Blur	36
3.3 Otsu Thresholding	37
3.4 K-means Clustering	38
3.5 Machine Learning.....	39
3.6 Artificial Neural Networks.....	39
3.7 Common CNN Layers	40
3.8 Network Optimization	42
3.9 Evaluation Metrics	46
4. Optical Flow Based Image Registration	47

4.1 Optical Flow	47
4.2 Dataset	49
4.3 Proposed Methodology	50
4.4 Results	53
5. CD3 Region Prediction on H&E Images	58
5.1 Generative Adversarial Networks	58
5.2 Proposed Methodology	60
5.3 Results	64
6. Conclusions	69
6.1 Contributions	69
6.2 Limitations and Future Work	70
References	72
Appendix-1 Pre-processing Algorithms	81
Appendix-2 Deep Learning Algorithms	87

List of Tables

Table 1: Some CD antigens and its corresponding cells	18
Table 2: Cost of Biopsies	25
Table 3: Cost of Surgeries	25
Table 4: Cost of Treatments.....	25
Table 5 Dataset Details	50
Table 6: Sample Values of Pixels for Overlap Analysis: Raw Values	56
Table 7: Sample Values of Pixels for Overlap Analysis: Normalized Values	56
Table 8: Mean of Similarity Measures	57
Table 9: Statistical Data from Algorithms	65
Table 10: CD3 Negative Regions Examples	67
Table 11: CD3 Positive Regions Examples.....	68

List of Figures

Figure 1: Tumour Microenvironment	16
Figure 2: Slide Preparation Steps	21
Figure 3: Tissue Microarray of a Tonsil Tissue	22
Figure 4: TMA of Colorectal Cancer Related Tissues	22
Figure 5: H&E-stained Sample	23
Figure 6: IHC and IF Images Example	24
Figure 7: Example of Medical Image Registration	34
Figure 8: Feature Matching using SURF	35
Figure 9: Gaussian Blurring on an H&E image	36
Figure 10: Otsu Thresholding on Blue Nuclei from an H&E image	37
Figure 11: K-means Clustering Example	38
Figure 12: A typical Artificial Neural Network	40
Figure 13: Convolution Layer Operation.....	41
Figure 14: Zero Padding	41
Figure 15: Application of Strides in CNNs	42
Figure 16: Max-Pooling Operation.....	42
Figure 17: Tanh Activation Function	45
Figure 18: ReLU Activation Function.....	45
Figure 19: Sigmoid Activation Function	46
Figure 20: Working of TVL1 Optical Flow method	48
Figure 21: Discarded Image Example (Improper Staining)	49
Figure 22: Discarded Image Example (Poor Registration)	50
Figure 23: Shades of Blue with RGB values.....	51

Figure 24: Image Registration Flow	52
Figure 25: Proposed Image Registration Sample.....	54
Figure 26: Overlap of Masks Before and After Registration	55
Figure 27: Boxplot of Similarity Measures	57
Figure 28: U-net Generator Structure.....	59
Figure 29: Process Workflow	61
Figure 30: Deep Learning Process Flow	62
Figure 31: Ground Truth Image Examples	63
Figure 32: Whole TMA Prediction Example	66

Chapter One

Introduction

1.1 Definition of Cancer and Related Statistics

The National Cancer Institute in the United States defines cancer as a combination of diseases in which malignant cells multiply uncontrollably and invade nearby tissues. The American Society of Clinical Oncology explains it as a group of more than hundred diseases which could occur anywhere in the body. When such a growth occurs and forms into a solid tissue, a tumour or neoplasm is said to be formed [1,2]. Some of these tumours will not affect neighbouring cells while some do, these are known as benign and malignant tumours, respectively. Benign tumours could potentially be harmful since large masses in areas like lungs or fibroids could create structural difficulties due to its size and cause pain. When tumours divide and spread to other sides via the blood stream, a process known as metastasis takes place, hence these tumours are known to be malignant in nature [3]. Cancers can broadly be divided into the following types:

1. Carcinoma: The most prevalent form of cancer, solid tumours are formed on the surface of tissues or skin above internal organs. Some examples of these type of cancers occur in the breast, lungs, or the prostate gland.
2. Sarcoma: This type of cancer occurs in the support structures of the body like bones, joints, or cartilages.
3. Leukaemia: This type of cancer affects healthy blood cells causing uncontrollable growth.
4. Lymphoma: Areas like the spleen and lymph nodes are affected which are essential components of the immune system.

GLOBOCON (Global Cancer Incidence, Mortality and Prevalence) from the World Health Organisation (WHO) provides useful statistics for cancer. It seems to be the leading cause of death partly due to advancements in other medical fields to prevent conditions like stroke. According to the statistics, Asia accounts for close to half of the cases and more than half of cancer related deaths. This is because close to sixty percent of the global population lives in these areas. Furthermore, female breast cancer, followed by lung and colorectal cancers are the most diagnosed cancer types. However, lung cancer-based mortalities are highest amongst other types. Not only is lung cancer the most found cancer type amongst men, but it is also the cause of most deaths occurring from cancer. In contrast, breast cancer is the most common type of cancer amongst females with most cancer related mortalities occurring due to it. Moreover the rates of breast cancer incidences surpass that of lung cancer, making it

the most diagnosed cancer type in the world [4]. In Singapore, Cancer is the leading cause of death and at least twenty thousand succumb to this illness [5].

Using data from the National Cancer Database (NCDB), the following treatment modalities are currently in use:

1. Surgery: Procedures involving partial or complete removal of affected regions e.g.: mastectomy which is the complete removal of breast tissue.
2. Radiation therapy: Also known as radiotherapy, it is a treatment modality where ionizing radiation is used to treat malignant cells, e.g. teletherapy which involves emission of radiation beams externally or brachytherapy in which a radioactive material is placed close to the tumour [6].
3. Chemotherapy: This therapy involves the usage of drugs to combat cancer, e.g. alkylating agents that affect DNA or antibiotics that affect the enzymes responsible for the replication of DNA [7].
4. Targeted therapy: Instead of targeting all uncontrollably dividing cells as in chemotherapy, smaller molecules, or monoclonal antibodies (mAbs) are used to provide “personalized” medication as specific targets are being identified [8].
5. Hormonal therapy: For cancers like oestrogen-receptor positive breast cancer, introduction of synthetic hormones to disrupt natural secretion of hormones is done in hormonal therapy [9].
6. Immunotherapy: The body’s innate immune system is employed to impede the progress of cancer. Some immunotherapy-based drugs can be combined with therapies like radiotherapy to achieve something close to a “cure” for cancer, e.g.: ipilimumab targeting cytotoxic T lymphocytes antigens (CTLA-4) or pembrolizumab which target Programmed Cell Death protein -1 (PD-1) [10].

Based on the aforementioned statistical information on the most prevalent cancer types, namely breast, colorectal and lung cancers the following are some statistics on their treatments based on different stages of progression [11]:

1. Breast cancer:
 - a. Stage 1 and 2: Breast conservation surgery with adjuvant radiation therapy (~50%) or mastectomy (~35%)
 - b. Stage 3: Mastectomy (~70%) and adjuvant chemotherapy
 - c. Stage 4: Radiation and/or chemotherapy (~60%), no treatment (~25%) or hormonal therapy
 - i. Increased use of hormonal therapy observed with patients having hormone receptor-positive tumours (~80%)
 - ii. Triple-negative tumours or TNBC treatment statistics are relatively poorer and are usually associated with chemotherapy. However, some immunotherapy drug and chemotherapy-based

treatment is under investigation. Nevertheless, this treatment seems to be effective only to a smaller population due to the heterogenous nature of molecular profiles.

2. Colon cancer:

- a. Stage 1 and 2: Colectomy without chemotherapy (~85%). Some stage 2 patients receive chemotherapy.
- b. Stage 3: adjuvant chemotherapy (~65%)
- c. Stage 4: Radiation and/or chemotherapy (29%) with surgery (~36%), colostomy (12%). Immunotherapy based on tumour characteristics could be used here.

3. Rectal cancer:

- a. Stage 1: Proctectomy/Proctocolectomy (~60%); Some stage 1 patients receive neoadjuvant radiation or chemotherapy.
- b. Stage 2 and 3: neoadjuvant chemoradiotherapy and surgery (~60%)
- c. Stage 4: Radiation and/or chemotherapy (52%) with surgery (~26%), colostomy (12%); Immunotherapy based on tumour characteristics could be used here.

4. Lung cancer:

- a. Stage 1 and 2: Surgery (~55%)
 - i. Types of surgery include lobectomy, pneumonectomy, or sleeve resection.
 - ii. Stage 3: adjuvant chemotherapy (~45%)
- b. Stage 4: Chemotherapy only (25%) or with radiotherapy (~26%), with a small percentage opting for immunotherapy (12%)
 - i. Lung cancer are of two types, small cell (SCLC) or non-small cell (NSCLC) lung cancer. The treatments statistics provided above are that of NSCLC since it accounts for (~83%) of overall cases as opposed to SCLC (~13%)
 - ii. SCLC cases are mostly treated with chemotherapy.

1.2 Tumour Micro-Environments

Tumour Micro-Environments (TMEs) provide knowledge about the intricate interaction of host and tumour cells, namely the formation and movement of the latter in the micro-environment, resulting in degradation of tissue integrity and carcinogenesis. A TME also includes blood vessels, signalling molecules like cytokines or galectins, non-malignant cells as well as a complex network of macromolecules arranged per cell or tissue known as the extracellular matrix (ECM) [12–15]. Figure 1 is a representation of a typical tumour micro-environment.

Figure 1: Tumour Microenvironment [12]

It is found the non-malignant cells in the TME, because of complex signalling networks ultimately promote metastasis. A definition of some of the components visible in the figure are given as follows:

1. Endothelial cells (EC): The general function of endothelial cells is to promote angiogenesis or the formation of blood vessels, which promote growth of cells, in this case the tumour cells [16].
2. Pericytes: Essential to endothelial cell growth, it secretes vascular endothelial growth factors (VEGF) to promote growth and protection from anti-angiogenesis [17].
3. Tumour Associated Macrophages (TAM): These macrophages are commonly found deeper in the in the TME and are associated with the promotion of angiogenesis and metastasis [18,19]
4. Cancer Associated Fibroblasts (CAF): Spindle-shaped fibroblasts, which are present superficially compared to TAMs usually assist in wound healing. However, CAFs tend to promote metastasis by affecting the ECM, drug delivery patterns and recently, even the immune response [19,20]
5. Adipocytes: These cells are responsible for energy regulation and storage. However, in the case of TMEs, they are also responsible in the secretion of hormones and growth factors amongst other items. Furthermore, as breasts mostly contain white adipose tissue, it provides fatty acids by the means of lipolysis (breakdown of lipid stores) useful for the TMEs [21].
6. Dendritic cells: Antigens are molecular structures which usually trigger an immune response. These antigens are captured by dendritic cells and are

presented to T cells which have an adaptive immune response. In the case of TMEs, instead of inhibiting its functioning, dendritic cells are modified to allow its presence and even block any immune response against it [21].

7. Apoptotic cells: These cells take part in a process called as apoptosis which is a type of cellular death process. Due to TMEs affecting signals needed for proper functioning of these cells amongst other factors, less to no apoptosis occurs [22].
8. Cell-free DNA: This type of circulating DNA is due to cellular death or secretions from cancer cells which are present in larger numbers as compared to healthy individuals. A small percentage of this, called as circulating tumour DNA (ctDNA) is being studied as potential biomarkers for cancer diagnosis or prognosis[23].
9. Natural killing cells: As the name suggests, these cells focus on killing infectious and tumorous cells, however these are less effective inside the TME [21].
10. Exosomes: In TMEs, its primary function is to communicate between stromal (ECs, Fibroblasts etc.) cells and other cancer cells to promote metastasis, inflammation amongst other processes.
11. T-cells: T-cells are special types of immune cells that mostly originate from the Thymus gland which is a part of the lymphatic system. There are two types of T-cells, the Th or T helper cells and the Tc or the T cytotoxic cells [24]. The former helps other immune cells like B cells or Tc cells itself to promote secretion of antibodies for an immune response while the latter, based on antigens targets cancerous cells for destruction. Therefore, the presence of either of these cells mean a positive prognosis for cancer. Tc cells, apart from antibody production also hinder angiogenesis via interferon gamma secretion. Furthermore, each T cell has its own receptor (T cell receptor or TCR) for specific antigen identification. Moreover, there are regulatory T cells (Tregs) which prevent autoimmune or inflammatory responses. However, these cells can help in tumour progression as they could also prevent the immune response against tumours [21]. Therefore, based on its relative location in the TME, T cells fall into three categories:
 - a. Immune infiltrated: Equally scattered T cells throughout the tumour signalling an active response.
 - b. Immune excluded: T cells at the periphery which have not yet infiltrated the TME.
 - c. Immune silent: T cells which have not at all engaged with the tumour, citing no immune response.
12. B-cells: These cells originate from the bone marrow and are found in the lymph nodes close to the TME. Although these are less in number compared to T cells, its association with T cells is a growing topic of interest as it is found that it is a positive prognostic marker primarily in breast and ovarian cancer. B cells help

in T cells by presenting antigens to act on and also help in the production and propagation of cytokines. In contrast, in the cases of prostate or bladder cancer, its presence usually leads to a poor prognostic result. Furthermore, like Treg cells, regulatory B cells may inadvertently help in tumour progression by reducing the immune response of immune cells like macrophages or Tc cells [21].

1.3 Clusters of Differentiation (CD)

The aforementioned components of a typical tumour microenvironment amongst various other cells contain certain molecules on their surface which eventually help in differentiation, with the help of monoclonal antibodies which are antibodies developed in a laboratory by exposing white blood cells to certain viral proteins. Since most of the components in consideration are leukocytes, anti-leukocyte monoclonal antibodies (mAbs) are employed. Therefore, different antigens react with specific mAbs which helps in identification of different leukocytes and hence it aids in cancer diagnosis and prognosis [25,26]. Table 1 provides a list of certain CD antigens useful for determining the components in a TME, extracted from over five hundred unique CD antigens found. Only antigens with known function or suitability in detecting certain cell types have been mentioned. Nevertheless, many antigens with suitable cellular expressions can be used to mark some specific types of cells.

Table 1: Some CD antigens and its corresponding cells

Cells	CD antigens
T cells	<p>CD2: Original markers for T cells</p> <p>CD3: Best for T lineage cells</p> <p>CD4: Th cell marker</p> <p>CD7: T cell ALL (Acute Lymphoblastic Leukaemia) marker, positive cells interact with B cells for autoimmune disease avoidance</p> <p>CD8: Tc cell marker</p> <p>CD27: Naïve T cell marker</p> <p>CD43: Helps in identifying malignant and normal T cells</p> <p>CD45RO: Memory T cell identifier</p> <p>CD60: Positive cells T cells act as helper cells to B cells</p> <p>CD69: Early T cell activation marker</p> <p>CD70: T cell activation helper</p>

	CD96: T cell ALL marker CD127: CD4 ⁺ CD25 ⁺ CD127 ⁻ T cells are associated with Tregs CD165: Found in conjunction with T cell ALL
B cells	CD19: Major B lineage cells marker CD20: B cell activation/propagation/differentiation CD22: Mature B cells malignancy diagnosis and experimental therapy indicator CD24: B cell development indicator CD37: Healthy and malignant B cell marker CD75: segregation amongst various malignant B cell types CD179a: Pre and pro B cell markers CD256: B cell development indicator CD257: Promotes B cell survival and activation, potential treatment target for B cell tumours CD307: Potential Immunotherapy marker for B cell leukaemia CD317: Late-stage B cell maturation marker
NK cells	CD16: NK cell marker CD56: NK cell marker, subsets based on CD56 expression CD57: NK cell marker CD335: Specific NK cell marker, expression density linked with NK cell cytolytic activity
Dendritic cells	CD83: Dendritic cell marker CD301: Immature dendritic cell marker
Macrophages	CD11b: Macrophage marker

1.4 Cancer Diagnosis

To examine the progress of cancer, imaging techniques like CT scans or MRIs are used. Another method to track its progress is to use a specimen of the tissue of the affected region and examine it using microscopes. This is also known as a biopsy. It can be done by the following methods:

1. Invasive Biopsies: Surgical procedures are used to remove required tissues from certain areas, like lungs.

- a. Incisional Biopsy: Part of the tumour and that of an adjacent tissue is removed for examination [27].
 - b. Excisional Biopsy: If the tumours are small enough and not too malignant, then it is completely removed for examination [27].
- 2. Minimally Invasive Biopsies: Procedures involving no surgical procedures, ensuring reduced trauma while obtaining samples.
 - a. Needle Biopsy: A minimally invasive technique employed if the usual methods cannot be used. Based on physical aspects like needle diameters or even the size and location of the tumours and the need of imaging guidance, this type of biopsy can further be divided into fine needle aspiration or core needle biopsies [27,28]. This method is prevalent for breast cancer patients.
 - b. Endoscopic Biopsy: Using endoscopes, tissues needed for examination can be removed without the need of surgical procedures [29]. This is used for colorectal cancer patients.
 - c. Skin Biopsy: If the needed specimens are accessible from the periphery, that is the skin, it is possible to use remove specimens using sterilized punches (punch biopsy) or scalpels [30]. This method is mainly used for skin cancer patients.
 - d. Liquid Biopsy: With advances in studies to interpret cfDNA and ctDNA, present in tumours has enabled the use of liquid biopsies. In this technique, the previously mentioned components and more can be derived directly from blood samples [31]. This method is not as widely used as the others however it is being tested as a potential alternative for invasive biopsies.

Preparation of slides by staining the obtained tissues and observations using different types of microscopes are used to determine the progress of cancer.

1.5 Histopathology and Cancer

Histology is defined as the study of the anatomy of the tissues and individual cells of an organism at a microscopic level. Several articles explain histological images of tissues to be the “gold” standard of cancer diagnosis, since most changes happen at the molecular level which could be easily understood under a microscope. Furthermore, since these processes have been in use for decades, there is also some form of standardization in every step. Histopathology is a subset of histology whereby biopsies are examined for the sole purpose of disease location and classification, in this case cancer [32,33]. Figure 2 summarizes the various steps involved in creating an observable slide of a tissue biopsy.

Figure 2: Slide Preparation Steps

For microscopic observation, either light-based or electron microscopes are used. In the case of light-based microscopes, a system consisting of lighting and imaging subsystems are used. In contrast, electron microscopes use accelerated electrons, and these electrons form an image using the diffracted electrons. These microscopes are expensive compared to light-based microscopes due to different sample preparation techniques, but resolutions of up to a few nanometres can be achieved and all optical issues of image-based microscopes are absent.

Image-based microscopes are further divided into fluorescence, confocal and multispectral microscopes, while electron microscopes are divided into transmission and scanning electron microscopes (TEMs and SEMs).

Pathologists nowadays do not have to routinely assess the tissue sections using microscopes, instead using specialized equipment, they can generate Whole-slide images (WSI). A comprehensive study conducted by Mukhopadhyay et. al. using a wide variety of tissues showed that there was not a lot of difference between the conventional glass slides and the WSIs, thereby calling WSIs “non-inferior” to conventional glass slides [34]. Figure 3 depicts a typical whole tissue section used for studies, which could not only result in gigabytes of data to process at a time for computer-aided diagnosis (CAD) methods but are also possible only via very expensive machines due to its sheer size. Hence, tissue microarrays (TMAs) are employed to address these issues.

Figure 3: Whole Slide Image of a Tonsil Tissue [35]

The potential use of TMAs was described in 1987, however only after a decade, it was possible to use it due to the development of a device which could ensure easy production of such samples. This was possible by using a special punch to extract important regions marked by pathologists. As these punches are cylindrical in nature, round blocks are procured. Due to these arrays being smaller in size and comparatively less in quantity as opposed to WSIs, multiple patients' data can be assessed on the same slide, each placed side-by-side. Additionally, there is also a possibility to assess the multiple stages of cancer progression as only regions of interest from each stage can now be placed next to each other. Furthermore, an individual image from a TMA has a much lesser size as opposed to a WSI. However, owing to the heterogeneity of tumours, a relatively large number of specimens need to be used to better represent the tumour accurately [36,37]. Figure 4 depicts a typical TMA slide.

Figure 4: TMA of Colorectal Cancer Related Tissues

For several years, Haematoxylin and Eosin (H&E) based stains are used for analysis. Haematoxylin is deep blue purple in nature and stains nucleic acids via an unknown process. Eosin stains all types of protein and is pink in colour. Therefore, in any tissue, the blue stains would correspond to nuclei while the variation of pink stains would correspond to the cytoplasm and the ECM [38]. Figure 5 depicts a part of a typical H&E-stained TMA section.

Figure 5: H&E-stained Sample

Another method to study tissue biopsies is by using mAbs or polyclonal antibodies as stains to check the given antigens' distribution, especially those of CD markers which accumulate based on antigen-antibody reactions. Apart from oncology, it could also be used to diagnose brain trauma and muscle diseases amongst other applications [39]. However, a more comprehensive analysis could be done with the help of procuring both H&E and IHC images. Figure shows a typical IHC image.

Typical IHC-stained tissues are viewed under a brightfield microscope, however if viewed by a fluorescence microscope, a special type of visualization known as immunofluorescence is obtained. In contrast with IHC, the procured IF images have better amplification of signals as well as improved sensitivity. Furthermore, IHC images which can show information of more than one antigen, as opposed to one, called as multiplex IHC or IF images. A possible method of doing this without using sophisticated machines would be to stain consecutive sections with different stains. Nevertheless, such a technique cannot fathom the possibilities of specialized machines since cells have limited area, thus limiting the number of stains. Moreover, the lack of excess tissues renders this moot compared to staining the same cell with different antigens. Figure 6a and b show typical brightfield mIHC images, while Figure 6c and Figure 6d depict the same TMA section segment, but the former depicts multiple channels, that of CD3 (green) and FOXP3 regions (pink), while the latter is a single channel, of the FOXP3 positive regions [40].

Figure 6: IHC and IF Images Example, a) mIHC sample 1; b) mIHC sample 2; c) mIF Sample; d) Single Channel mIF Sample [40]

PD-1 and PD-L1 expressions are found to be present on tumour infiltrating CD8+ T cells, which are related with the expressions of CD8, CD3 and fork-head box P3 proteins (FOXP3). These expressions help, individually or in combination, in deriving useful prognostic inferences. Hence, using mIHC or mIF images prove to be an advantage during diagnosis. Furthermore, PD-1 expressions also have their use specifically for immunotherapy-based treatments. Additionally, T and B cells in proximity, as observed mIF images was inferred as enhanced immune activation leading to a better prognosis for hepatocellular carcinoma (HCC, liver cancer) [41].

1.6 Cancer Treatment Expenses

The use of newer types of diagnosis methods and new treatment techniques would add up to the usual costs involved for cancer treatment. To understand how expensive some of the abovementioned techniques could be in a country like Singapore, some relevant articles are used [42,43]. Table 2 is for biopsies, Table 3 is for surgeries and Table 4 is for treatment.

Table 2: Cost of Biopsies

Cancer Type	Cost (Singapore Dollars)
Breast	915-6275
Prostate	939-6067
Colorectal	1160-3730
Skin	492-3180
Average	877- 4813

Table 3: Cost of Surgeries

Cancer Type	Cost (Singapore Dollars)
Breast	1777-18525
Prostate	2030-53326
Colorectal	4177-23041
Lung	2426-25240
Average	2603- 30333

Table 4: Cost of Treatments

Cancer Type	Cost (Singapore Dollars)
Chemotherapy	1500 per cycle
Radiation Therapy	13000-30000 for a six-week, 15–30-minute session
Immunotherapy	9000

The expenses mentioned in the tables above are in ranges and are also differing based on cancer type due to the following reasons:

1. Availability of subsidies
2. Convenience of opting private healthcare facilities instead of public healthcare facilities.
3. Stages of cancer and the difficulties associated in obtaining biopsies and performing surgeries.

Use of newer and improved testing techniques or surgeries reduce costs involved, for example, instead of performing multiple conventional IHC staining, automated multiplex IHC solutions can be used, which amount to 2000 USD for fifty runs, which is 40 USD per run, using the same slide. Furthermore, the possibility of loaning such devices nullifies the need of a large upfront investment [41].

1.7 Objective and Scope

The core objective of this project is to use machine learning algorithms to develop a pipeline to obtain multiplex Immunofluorescence image data directly from Haematoxylin & Eosin images only. This is because irrespective of staining methodology, the tissue being studied is the same and there would be some details in the Haematoxylin & Eosin image which could be easy to discern only with the help of special deep-learning algorithms. Additionally, these algorithms would unravel a lot of hidden data contained in the inexpensive Haematoxylin & Eosin image and help reduce the overall costs of cancer treatment in general.

The scope of the project is as follows:

1. To use a computationally less expensive intensity-based registration algorithm to align Haematoxylin & Eosin and multiplex Immunofluorescence images.
2. To predict Cluster of Differentiation (CD) 3 regions as in multiplex Immunofluorescence images using Haematoxylin & Eosin images with the help of unsupervised machine learning algorithms

1.8 Thesis Organization

After the introduction chapter, there are three chapters. Chapter two is about the literature survey about concepts used in the subsequent chapters. Chapter three deals with related concepts needed to understand chapters four and five, which is about optical flow-based image registration and generative adversarial networks-based generation of virtual CD3 regions on H&E images. The last chapter is for contributions and future work.

There are two appendices, one for image registration scripts and the other for deep learning scripts.

Chapter Two

Literature Survey

2.1. Image Registration

Cooper et.al. in their article use feature-based registration to align H&E and CD3 based IHC WSIs. High-level features are aligned and techniques like rigid initialization, meaning model rigid transformation matrices are generated for the x and y axes as well as the rotation parameters and non-rigid feature matching to reduce strict matching and promote better alignment. For comparison, a ratio shown in equation (1) between a part of the superimposed area of the aligned moving image is compared with the area of the initial moving image, in this case the IHC image. The authors also claim that since different staining do not provide a visually similar image, similarity measures like mutual information may fail to work [44].

$$r = \frac{\text{Area}(T(S_{\text{IHC}}) \cap S_{\text{H\&E}})}{\text{Area}(S_{\text{IHC}})} \quad (1)$$

Keikhosravi et. al. in their article use intensity-based registration to align H&E and second harmonic generation (SHG) images. Intensity based registration approaches are opted as opposed to feature-based registration approaches like SIFT because intensity-based registration methods are independent of contrast mechanisms which differ with multimodal images. However, feature-based recognition methods, even with enough pre-processing seem to perform poorly with multi-modal images due to edge and texture distortion. The mutual information similarity function was applied between the images was used and its maximization provided with the means of effective registration. To determine the level of success, mean absolute error was used [45].

Peñaranda et.al. in their article explore the multimodal registration of H&E images and Fourier Transform Infrared Spectroscopy (FTIR) images. A combination of feature-based registration methods like SIFT and RANSAC have been used along with intensity-based registration method for improved alignment [46]. For computing similarity, mutual information and correlation ratio (CR) are used. Furthermore, a root-mean square error is also calculated for each mode of registration for evaluation. The equation of correlation ratio is a given by equation (2).

$$\text{CR} = 1 - \frac{1}{N\sigma^2} \sum_i N_i \sigma_i^2 \quad (2)$$

Where N is the number of pixels in the overlapping region Ω between the images. The other parameters are given by:

$$\sigma = \frac{1}{N} \sum_{x \in \Omega} T(x)^2 - m^2 \quad (3)$$

$$\sigma_i = \frac{1}{N} \sum_{x \in \Omega} T(x)^2 - m_i^2 \quad (4)$$

$$m = \frac{1}{N} \sum_{x \in \Omega} T(x) \quad (5)$$

$$m_i = \frac{1}{N_i} \sum_{x \in \Omega_i} T(x) \quad (6)$$

The subset i is used to denote a region of pixels having the same intensity level i . CR takes a value between zero and one. Ideally, a perfect alignment of images would have CR equal to one.

Jiang et.al. [47] in their article register H&E and IHC WSIs using the following steps:

1. Fourier Transform based Image Registration: To improve low resolution WSI registration, Fourier transform functions are used solely to get the appropriate transformations of position and orientation, since the images used did not need any scaling or affine transformations.
2. Kernel Density Estimator: A kernel density estimation (KDE) algorithm was used to further improve the Fourier Transform based registration approach. The KDE algorithm is a non-parametric method to predict the probability density function of a random variable, in this case the position and orientation of images. This method helps in reducing errors in smaller regions of images due to limited texture information by decreasing its contribution to the overall offsets and angles.
3. Linear Relationships: The concept of hierarchical resolution registration was used since WSIs at different resolutions differ in texture details and hence image pairs at unequal resolutions cannot be paired properly. A cost function involving a linear regression equation is formed, where the cost is minimized to achieve an optimal slope and hence the needed offsets to be applied on imaging having different resolutions.

These steps were performed to enhance the alignment of patch based WSIs, which are parts of large WSIs which are less computationally expensive to perform tasks on but suffer from texture inconsistencies.

Wang et.al. in their article also used three steps to develop an automated and quick registration method [48]. The steps are as follows:

1. Data Normalization: The RGB colour space is used where histogram equalization is performed specifically in the red and blue channels of the H&E image separately to properly distinguish regions. Furthermore, a vector-based

representation of the RGB colour space is also used to effectively separate the haematoxylin and eosin stains.

2. Sparse Approximation: An approximated registration process is opted where features selected by algorithms like random sample consensus (RANSAC) are used to derive a “sparse” representation of the images and the smallest Euclidean distance is found between them for the right transformation. This results in an escape from local minimum traps occurring mainly due to the varying textures of the given images as well as faster registration times.
3. Area-Based Direct Matching: Using parts of the bi-directional elastic b-spline model, an energy function based on terms like divergence and curl is minimized to calculate a term known as the elastic deformation field.

Hermann et.al. in their article also used the concept of elastic deformation field along with simple rigid transformations to align various slides of H&E and IHC images to be viewed in a three-dimensional environment [49]. Furthermore, a similar article by Lotz et.al. focuses on discretizing the deformation field [50].

Jackson et.al. in their article used a combination of rigid and non-rigid transformation to align WSIs of H&E and SOX10 based IHC images. ROIs were marked with the help of trained pathologists, on which rigid affine transformation was performed at a higher resolution while non-rigid transformation at a lower registration was performed at a lower resolution [51].

A recent and the closest articles on which most concepts of this thesis are based are that of Burlingame et.al., due to the similarities in the dataset used and its overall machine learning objective [52,53]. The DAPI (4',6-diamidino-2-phenylindole) stained regions of mIF images and the haematoxylin regions in H&E image correspond to the same regions, that of nuclei. Hence the transformations of DAPI regions, with respect to H&E image can be easily derived and then can also be applied to other channels of the mIF image to fully align the all the antibody stains. The registration process is as follows:

1. Grayscale conversion of images.
2. Creation of binary masks and then its complement to isolate H&E nuclei and the DAPI regions.
3. Use of SURF to align these images to obtain transformation matrices.
4. Application of transformation matrix on DAPI as well as other regions.

2.2. Medical Imaging Using Deep Learning

Several neural networks are used in many areas of medical imaging to bring about useful information. Scheduling of radiation therapy-based appointments based on

patient specific dosage is possible because of such networks. Moreover, there are studies where neural networks are used to interpret medical images like that of a pathological/radiological report using aspects of natural language processing.

For brain related images, CNNs have been used on different types of medical images like Computed Tomography (CT) and Magnetic Resonance Imaging (MRI) images to classify conditions like Alzheimer's disease or Schizophrenia. Moreover, some articles also focus on brain tumour segmentation and identification as well as advanced information like the prediction of survival or disease development [54]. The use of various, but similar loss functions was observed, e.g., the use of categorical cross-entropy for segmenting multiple regions of an image at once [55]. Furthermore, Yoo et.al. in their article improved the synthesis of reference images to enhance the information density of binary masks, which were otherwise blank, using Euclidean distance transforms, but in a three-dimensional space [56].

For eye related diseases, Fundus Imaging (CFI) is used for ophthalmological examination and the use of CNNs help understand blood vessel segmentation and detection of serious diseases like Glaucoma. An article by Gao et.al. used pre-trained filters, based on popular algorithms like k-means clustering and gaussian mixtures to effectively extract features, before subjecting it to CNNs and Recursive Neural Networks (RNNs) which are networks that work by repeatedly using the same parts of the parent network in a tree-like structure [57]. Additionally, Schlegl et.al. in their article also used semantic information based on medical reports on patients as an alternative to manual annotation on Optical Coherence Tomography [58].

Chest X-rays and CT scans help doctors predict Tuberculosis (TB) and other important aspects of the chest area like bone images [54,59]. A recent article by Chaunzwa et.al. used CNN-based feature maps instead of information from conventional classifiers for improved feature extraction [60].

For breast related diseases, mainly breast cancers, CNNs have been used on data obtained via mammography and ultrasound. Objectives include mass detection and localization, tissue classification and even cardiovascular disease classification based on calcification in vessels [54].

MRI imaging is the most common imaging method used for cardiac diseases and CNNs are used to detect and segment ventricles of the heart. Ultrasound (US) data coupled with CNNs can be used to detect and segment three-dimensional aortic valves [54]. Wu et.al. in their article were able to diagnose breast arterial calcification (BAC), an indicator of coronary artery disease using a custom CNN with an area-based receiver operating characteristic metric. However, this metric was applied only on patches containing some BAC, as images with no traces of BAC would give odd scores [61].

For the abdomen area, colonoscopy data with CNNs can help in polyp detection. Additionally, CT, MRI or US data can be used lesion detection in the case of livers, kidneys and bladders [54]. One article by Pogorelov et.al. proposed a GAN (Generative Adversarial Networks) -based approach to detect polyps in colonoscopy videos as a promising alternative to other deep learning paradigms like YOLO v2. The GAN paradigm was derived from a GAN paradigm developed for vessel segmentation for fundus images. The algorithm was able to detect polyps automatically at various levels, i.e., at the pixel-level, block-level and even at a frame level, which helped in deriving key information out of large and difficult datasets [62].

In the musculoskeletal regions, MRI data with CNNs can help in the segmentation of knee cartilages or the localization of vertebrae. Moreover, X-ray data can be used for bone age assessment, osteoarthritis scoring or even real-time arthroplasty kinematics registration either in two or three dimensions [54].

Pathology is another field where deep learning architectures are widely in use, for example mitosis detection, bacterial colony counting and detection of elements like mitochondria, TIL, cells and nuclei in images like H&E, EM or IHC images [54]. However, Lahiani et. al. in their article achieved the opposite of the objective of this thesis, i.e., they derived H&E images using mIF images using least squares optimization and manipulations in the optical density space without the aid of any deep learning method, which further proves that mIF images contain vastly superior information compared to H&E images and some transformations if applied, can convert images to one another [63].

Some articles use pre-trained networks like VGG16, YOLO, Inception etc. which are trained on several common images, hence structural features can be better analysed by these networks and by replacing the last layer of these networks with an additional layer(s), the pre-trained network can be used for newer tasks, mainly in classification and the weights are assigned only in the last few layers after training. This type of deep learning method is called as transfer learning. Moreover, the use of Gradient Weighted Class Activation Mapping (GradCAM) has helped to improve the interpretability of deep learning paradigms, e.g., showing how one algorithm is different from another even while each provide the same result or at least give a good explanation, in terms of heatmaps as to why a particular result has been obtained [64–67].

The main application of CNNs that is of special interest is its use in multimodal image synthesis or the generation of “virtual” images of one imaging modality using data available solely from another modality [68]. Nie et.al. for example, derived CT images using MRI data using GANs [69]. In another article by Nie et.al., 7T MRI images were generated using 3T MRI images [70]. Chatsias et.al. focus on another type of GAN called as cycleGAN to generate CT images from MRI images particularly for cardiac cases [71]. Costa et.al. in their article use GANs to create retinal images of various

colours [72]. Khosravi et.al in their article were able to couple MRI and H&E images of the prostate to evaluate the risk of cancer based on common features observed in both modalities derived using Inception V1. Furthermore, GradCAM was also employed to give a better explanation of the results obtained [73].

Burlingame et.al., in their articles use H&E images to derive mIF data using a version of GAN combined with another deep learning approach called as label-free determination, mainly images with stains like pan-CK (Cyto-keratin) and α -SMA (Smooth Muscle Actin). A Variational Autoencoder was used after image registration to reduce the overwhelming number of features to a feature vector of size 16. This step helped in segregating images based on the level of representation of the important features and redundancy [52,53].

GANs using histopathological images can be used to generate realistic images of a given modality which could be used for training networks for the previously mentioned applications when there is a lack of data, like the works of Levine et.al. and Quiros et.al.[74,75]. Furthermore, virtual staining of a given tissue sample can be performed, reducing the amount of tissue needed for analysis, like the work of Xu et.al. where they use additional loss functions like deterministic loss which helps in providing tissue subtypes and structural loss functions like photorealism loss and SSIM loss. Similarly, Rana et.al. in their article were able to stain unstained biopsy samples using a version of the Pix2Pix GAN model, which included Pearson's Correlation Coefficient as an additional loss function, to reduce tiling artifacts [76,77].

Chapter Three

AI Related Fundamentals

3.1. Image Registration

The images which are being used for analysis are mIF and H&E images of the same tissue section. However, due to the generation of separate WSI files of similar, not the same resolution and some alignment differences or even sample preparation procedures, the superimposition of the given images would not be fully aligned. Therefore, a pre-processing step known as image registration is done to align the given sets of images. Figure 9 is an example of medical image registration.

Figure 7: Example of Medical Image Registration, a) Before Registration; b) After Registration [78]

Image registration is done with a set of three components:

1. Fixed Image: This is the reference image where no transformations are performed.
2. Moving Image: This image is used for comparison with the fixed image and for eventual transformation.
3. Transformation: This function is applied on the moving image for it to be aligned to the fixed image.

While there are different types of image acquisition modes, like multi-temporal (image taken at different times) or multi-view (images taken at different orientation), the mode that is of particular interest is of multimodal image acquisition, where the same tissue is used in the same orientation, but different devices are used to capture different images, in this case the brightfield H&E-stained image versus the mIF image procured via a fluorescence microscope.

Image registration methods can be classified into the following types:

1. Feature based registration: Special features like points, lines, contours etc. are used to align images. This technique is used if the images have high detail. The steps involved in this method are feature detection, extraction, matching, transformation and finally resampling. Algorithms in use are SIFT (Scale Invariant Feature Transformation), SURF (Speeded Up Robust Features) etc. Figure x shows the feature matching process using SURF.

Figure 8: Feature Matching using SURF [79]

2. Intensity-based registration: No specific features are needed, rather similarity functions between the overlapping image pixels or comparison of optical flow vectors could be used. This technique is used when the given images are not rich in detail. One such function commonly used is mutual information. To understand the working of image registration, consider two variables A and B. The mutual information $I(A, B)$ is given by equation (7).

$$I(A, B) = H(A) + H(B) - H(A, B) \quad (7)$$

Where H is the entropy function given by equations (8) and (9)

$$H(A) = \sum_a -p_A(a) \log p_A(a) \quad (8)$$

$$H(A, B) = \sum_{a,b} -p_{A,B}(a, b) \log p_{A,B}(a, b) \quad (9)$$

Where p is the marginal probability mass function. For joint probability distributions, that is $p(a, b)$, a joint histogram matrix is calculated which is based on pixels of both images having their respective intensity values. Other functions include Fourier and correlation methods.

Based on staining, two types of image registration and transformation methods are applicable:

1. Rigid: This is performed in re-stained tissues due to lesser number of changes in positions and almost no change in orientation
2. Non-rigid: This is performed in serial sections due to changes in position and orientation.

3.2. Gaussian Blur

Gaussian blur is a method of removing noise or overlapping some regions of an image which would greatly improve in image registration. The “blurring” effect is obtained by performing a convolution operation on the images, which is a function applied on two functions leading to a third function which represents how one function affects the other [80]. The gaussian filter function, as seen in equation (10) is applied on the initial image via kernels which are usually oddly sized square matrices with the total sum of its elements being equal to one. Equation (11) represents a 5x5 gaussian kernel and Figure 9 shows its application. A kernel of size 55 is used to show an observable difference between the images.

$$G(x,y) = \frac{1}{2\pi\sigma^2} e^{-\frac{x^2+y^2}{2\sigma^2}} \quad (10)$$

$$\text{Gaussian 5x5 Kernel} = \frac{1}{273} \begin{bmatrix} 1 & 4 & 7 & 4 & 1 \\ 4 & 16 & 26 & 16 & 4 \\ 7 & 26 & 41 & 26 & 7 \\ 4 & 16 & 26 & 16 & 4 \\ 1 & 4 & 7 & 4 & 1 \end{bmatrix} \quad (11)$$

a.

b.

Figure 9: Gaussian Blurring on an H&E image, a) Before; b) After

3.3. Otsu Thresholding

Thresholding in image processing is a process in which the pixels having grey levels above a certain threshold are changed to a given value, mostly the highest value of the given level, in this case 255 and the remaining pixels are set to zero. This process helps in creating masks with pixels bearing only two distinct values which would help in easier registration. There are several thresholding methods based on what type of image related information they use, for example, histogram or entropy- based thresholding methods. However, there is a clustering-based thresholding method which frequently used developed by Nobuyuki Otsu [81,82]. As previously discussed in the previous K-means clustering sub-section, clustering can work easily with any dataset and dynamically provide the thresholding value needed. The process of thresholding is summarized in the following steps:

1. Normalization of the grey levels of the pixels.
 - a. The plotted histogram will now become a probability distribution.
 - b. Division of pixels into two classes
2. Due to normalization, the sum of probabilities of pixels belonging to one class would be a complement to that of the other class.
 - a. Calculation of within class and between class variances.
 - b. The addition of both these values would result in the total variance of levels.
3. The objective for proper thresholding is to then minimize the within class variance which would result in the increase in between class variance meaning clustering would be perfect.

Figure 10 is an example of the working of Otsu thresholding.

Figure 10: Otsu Thresholding on Blue Nuclei from an H&E image, a) Before Thresholding; b) After Thresholding

3.4. K-Means Clustering

Clustering is the method of grouping similar quantities into separate groups based on similarity called clusters. One such method to perform clustering is by the K-means method. This method employs the use of cluster centroids c to differentiate between data for a given k number of clusters [83]. Equation (12) shows the objective function that must be minimized to find the optimal coordinates of centroids. This function is the widely used Euclidean distance function between the individual points and the centroid which is arbitrarily chosen.

$$J = \sum_{j=1}^k \sum_{i=1}^n \left\| x_i^{(j)} - c_j \right\|^2 \quad (12)$$

The simplified algorithm for K-means clustering is as follows:

1. Choose K number of clusters.
2. Assign arbitrary values for the cluster centroids.
3. Assign the closest centroid to each data point.
4. Change the value of the centroid based on step 3.
5. Check if data points are close to the new centroids, else repeat from step 3.

The same algorithm can also be applied on RGB images where each pixel value is a row array with three columns with values of each element ranging from 0 to 255. Figure 1 shows a sample image from the dataset and its individual clusters. The value of k was 5 but the fifth cluster was a black image, hence omitted from the figure.

Figure 11: K-means Clustering Example a) Given Image; b) Cluster 1; c) Cluster 2; d) Cluster 3; e) Cluster 4

3.5. Machine Learning

Machine learning is the process by which a computer can learn and perform certain tasks without the need of explicit rules being programmed in it. Hence, via machine learning, algorithms can be generated from which predictions can be made which is mainly done by a series of transformations to the given dataset [68]. Machine learning paradigms can be divided into three main parts:

1. Supervised Learning: There is a separate training step before prediction in which samples from experts known as “ground truth” are provided to the algorithm to train and then come up with a set of rules to interpret it. Usually, two types of problems are solved using such paradigms:
 - a. Regression: Prediction of a certain dependent variable based on given independent variables, e.g., ambient temperature prediction based on parameters like relative humidity, luminosity etc.
 - b. Classification: Prediction of where a certain variable falls within a set of defined classes, e.g., if the given ambient temperature should be construed as being hot or cold.
2. Unsupervised Learning: In this case, no ground truth is provided. Hence the algorithm with no particular experience with the data would have to work with it. These types of problems are usually solved using such paradigms:
 - a. Clustering: Grouping of data based on similarities, e.g., colours of a given image.
 - b. Dimensionality Reduction: Calculation of the most important figures for a given dataset, e.g., reduction of 16-feature classification dataset into 3 important features for faster analysis.
3. Reinforcement Learning: In this method, effective penalization and reward measures are used to train the algorithm while letting it interact with the given data.

3.6. Artificial Neural Networks

A part of machine learning which is of great interest is that of artificial neural networks which are based on the complex connections and parallel processing capabilities of the human brain. A basic equation (13) is used to explain how the output is calculated.

Figure 12 shows a typical artificial neural network.

$$\text{Output} = \text{Weights} * \text{Input} + \text{Bias} \quad (13)$$

Figure 12: A typical Artificial Neural Network

The weights correspond to the strength of each connection from the input layer to the output layer of neurons while a bias is a random value being added to offset the given output and to ensure that there is some activation even if all inputs are zero. All layers of neurons between the input and the output are termed as “hidden layers”. If the number of hidden layers is more than one, then these networks are part of subset of neural network algorithms called as deep learning algorithms. Deep learning algorithms can be divided based on its inputs.

1. Typical Neural Networks: These networks take a one-dimensional vector as input.
2. Convolutional Neural Networks (CNNs): These take two- or three-dimensional vector data as inputs.

Grayscale images are two dimensional in nature while coloured images are three dimensional in nature, hence CNNs are of great importance for such cases. Like a typical neural network, there are the usual layers, but the hidden layer processes a convolution function on the input to transform the input [68,84].

3.7. Common CNN Layers

Most CNNs contain the following three layers:

1. Convolution Layer: A matrix much smaller than the given image (an array of numbers known as tensors) known as a kernel is used where element-wise multiplication is applied and the summed value is used to create a smaller matrix, hence reducing a large image a smaller matrix with supposed important features, therefore the resulting matrix is called as a feature map and the kernel can be called as a feature extractor. Typical sizes of kernels range from 3x3 and possibly to 7x7. Figure 13 shows a simplified process of the convolutional layer operation.

Figure 13: Convolution Layer Operation

- a. Figure 14 shows a case where the kernel size does not clearly match the size of the part of the matrix used, hence the remaining elements needed to equalize the shape for the convolution operation are made to be zero and this is termed as padding.

Figure 14: Zero Padding

- b. The degree of overlap between two kernel positions can be defined with the help of a parameter known as stride. If the value of stride is greater than one, then there will be an overlap between each kernel, as shown in Figure 15.

Figure 15: Application of Strides in CNNs

2. **Pooling Layer:** While convolution layers are layers in which the individual number of kernel elements are learnt, pooling is a computationally inexpensive method to reduce the dimensionality of data since it uses a ready-made kernel to extract the features, usually from the previous convolutional layer. Figure 16 shows an example of pooling.

Figure 16: Max-Pooling Operation

3. **Fully Connected Layer:** Usually present at the end of the network, this layer could be a one dimensional / flattened layer of nodes followed by some activation functions which could provide outputs like the probability of a pixel belonging to which class, which is essentially an image segmentation problem.

3.8. Network Optimization

There are two types of components in each network, they are termed as the learning parameters and hyperparameters. The former is what the networks try to learn from the given dataset, e.g., kernel elements in the convolutional layers, while the latter are

parameters which are initially set and are usually not changed, at least not during the training phase, e.g., stride and kernel sizes.

A common issue faced by neural networks in general are overfitting or extreme adaptation of the model to the given dataset to the extent that it performs badly with other datasets [68]. Hence to mitigate the possibility of overfitting, the following strategies are used:

1. Larger Datasets: Larger datasets reveal all possible variations possible in the dataset, hence it would help in better generalization for the CNN.
2. Batch Normalization: This step ensures that the into subsequent layers of a CNN would be normalized with the help of a batch normalization layer.
3. Dropouts: A very useful layer whose objective is to randomly ignore the contribution of certain layers to better regularize the data.

The learning strategy of a CNN takes place by two processes, forward propagation, and backward propagation. In the former step, the initially assigned parameters like kernel elements are used on the training dataset and its performance is observed and in the latter step, the parameters are updated to learn the dataset better. Forward propagation is carried out with the help of special loss function which basically compute the errors between the predicted and expected outputs. Examples of basic loss functions include mean squared error, usually used for regressions or cross entropy for classification. For cross entropy loss, something similar to equation (7) is used. Equation (14) shows the cross-entropy loss and equation (15) shows the equation for binary cross entropy loss which is used for binary classification [85].

$$L_{CE} = - \sum_{i=1}^n t_i \log(p_i) \quad (14)$$

Where t_i is the truth label and p_i is the probability.

When $n=2$, this loss is now called as binary cross-entropy.

$$L_{BCE} = - \sum_{i=1}^2 t_i \log(p_i) = -[t \log(p) + (1 - t) \log(1 - p)] \quad (15)$$

An average of values obtained by equation (15) on several data points is used to calculate the total loss.

For backward propagation, a method known as gradient descent is frequently used. A hyperparameter called learning rate is used with this which is basically the step size for learning, which means that a lower value will iteratively alter the values albeit at a slower pace, but a faster learning rate would do this faster but may lead to poor results. There are some learning rate optimization methods like using genetic algorithms to identify the best possible values for hyperparameters. Gradient descent modifies the weights of a network, hence following equation shows the working of this method:

$$\text{Modified Weight} = \text{Previous Weight} - \text{Learning rate} * \frac{\partial(\text{Loss function})}{\partial(\text{Weight})} \quad (16)$$

This approach for larger datasets will take time since the update is done after traversing through the whole dataset, hence the use of stochastic gradient descent (SGD) and a popular optimization algorithm called as Adam (Adaptive Moment) optimizer is used instead. In SGD, weights are calculated after each sample, that is, for a learning rate of η , if the usual gradient descent method is given by equation (17), the SGD formula is given by equation (18):

$$w_j = w + \Delta w_j \text{ where } \Delta w_j = \eta \sum_i (\text{target}^{(i)} - \text{output}^{(i)}) x_j^{(i)} \quad (17)$$

$$w_j = w + \Delta w_j \text{ where } \Delta w_j = \eta (\text{target}^{(i)} - \text{output}^{(i)}) x_j^{(i)} \quad (18)$$

The Adam algorithm for a given learning rate α and hyperparameters β_1 and β_2 uses the concept of moving averages to update parameters. Two moving averages, one of the SGD value itself and one of its squares are used, modified for bias and then used for updating [86]. Equations (19-22) is followed sequentially for the algorithm.

For a given SGD value g_t ,

$$\text{First Moment } m_t = \beta_1 * m_{t-1} + (1-\beta_1) * g_t \quad (19)$$

$$\text{Second Moment } v_t = \beta_2 * v_{t-1} + (1-\beta_2) * g_t^2 \quad (20)$$

$$\text{Corrected First Moment} = \hat{m}_t = \frac{m_t}{1-\beta_1^t} \quad (21)$$

$$\text{Corrected Second Moment} = \hat{v}_t = \frac{v_t}{1-\beta_2^t} \quad (22)$$

An activation function is an integral part of all neural networks since the output of each node and hence that of each layer is determined by this function. The following functions are used as activation functions:

1. Hyperbolic Tangent Function:

$$\tanh(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}} \quad (23)$$

Figure 17: Tanh Activation Function

2. Rectified Linear Activation Function:

$$\text{ReLU}(x) = \begin{cases} 0 & \text{if } x < 0 \\ x & \text{if } x \geq 0 \end{cases} \quad (24)$$

Figure 18: ReLU Activation Function

3. Sigmoid Activation Function:

$$S(x) = \frac{1}{1+e^{-x}} \quad (25)$$

Figure 19: Sigmoid Activation Function

3.9. Evaluation Metrics

Some metrics like peak signal to noise ratio (PSNR) as seen in equation (26), structural similarity index (SSIM) as seen in equation (27) and the ratio seen in equation (1) are used to determine if there is an improvement in overlapping before and after the proposed image registration method or deep learning methods.

$$\text{PSNR} =$$

$$10 \log_{10}(\text{MAX}_{\text{INTENSITY}}) - 10 \log_{10}\left(\frac{1}{m \times n} \sum_{i=0}^{m-1} \sum_{j=0}^{n-1} [I(i, j) - K(i, j)]^2\right) \quad (26)$$

Where I and K are the given images of size m x n.

$$\text{SSIM} = \frac{(2\mu_x\mu_y + c_1)(2\sigma_{xy} + c_2)}{(\mu_x^2 + \mu_y^2 + c_1)(\sigma_x^2 + \sigma_y^2 + c_2)} \quad (27)$$

Where μ are average values and σ standard deviation values of images x and y. σ_{xy} is the covariance of both images and c_1 and c_2 are variables used for stabilization of the output.

SSIM varies between 0 to 1, where a value of 1 indicates that the given images are totally similar. On the other hand, PSNR values could be in the range 10-40 and even higher, where the higher value indicates more similarity.

Chapter Four

Optical Flow Based Image Registration

4.1. Optical Flow

For the current dataset, feature-based registration methods provide poor results since as previously explained, there is an inconsistency with textures in TMAs. Therefore, another approach for image registration, which is not used before for pathological image registration, especially for mIF and H&E images is by using optical flow. Horn and Schunck in their article explain optical flow “as the distribution of apparent velocities of movement of brightness patterns in an image” [87]. Noe et.al. in their article explained the so-called “optical velocity” vectors as vectors that describe the direction of image intensity flow [88]. Optical flow is usually used for object tracking in videos, but it can be also used for individual images [89,90].

Optical flow is used to define the gradient of the grey values between two images which results in a displacement field vector given by [91]:

$$d(x, y) = \begin{pmatrix} u(x, y) \\ v(x, y) \end{pmatrix} \quad (28)$$

Where x and y are coordinates of an image pixel. The key component to form a vector field is to effectively quantify the brightness of each (x, y) point, in this case the intensity at the point is used. Therefore, given displacements u and v in the x and y direction respectively, to effectively equate two images, say A and B where A is the reference image and B is the moving image, such a transformation is expected:

$$I_B(x + u(x, y), y + v(x, y)) = I_A(x, y) \quad (29)$$

Concepts of vector algebra and advanced calculus concepts like weak forms of equations and partial differential equations are used to calculate the vector fields and hence the required transformation needed to warp the given image to the reference image. A summary of some of the methods are as follows:

1. Horn-Schunck method: It is assumed that the changes in brightness are consistent. Hence with the aid of equation (29), a penalized derivative is added to smoothen the vector field. Hence the resulting equation is globally minimized to help us attain the values of u and v . Therefore, for both terms, the least-squares minimization approach is used. Equation 28 is provided for reference [87,92,93].

$$\min_{u(x,y), v(x,y)} \left\{ \lambda \int |\nabla u(x, y)|^2 + |\nabla v(x, y)|^2 d\Omega + \int (I(x, y) - I(x + u(x, y), y + v(x, y)))^2 d\Omega \right\} \quad (30)$$

Where λ is needed for the penalized derivative term and both integrations are applied over the image Ω .

2. Iterative Lucas-Kanade method: As opposed to the global optimization approach, in this method, a local optimization method is opted, eliminating the need of the penalized derivative. Like the Horn-Schunck method, the least squares minimization approach is used. Equation (31) is provided for reference [88].

$$\min_{W(x,y;p)} \{ \int (I(x,y) - I(W(x,y;p)))^2 d\eta \} \quad (31)$$

Where η is subset of the image and W is the image warping function and p is the vector containing the warping parameters. Hence the objective is to optimize the warping function.

3. Total Variation (TV)- L^1 method: This is the method that is used for this study, as well as improvement on the Horn-Schunck method by changing the least squares optimization approach to an L1-norm, which is the absolute difference amongst the given elements. This results in a faster, yet stable iterative algorithm to obtain the needed vectors which can also with inconsistencies in brightness. Equation (32) is provided for reference [92,93].

$$\min_{u(x,y), v(x,y)} \left\{ \int |\nabla u(x,y)| + |\nabla v(x,y)| d\Omega + \lambda \int |I(x,y) - I(x + u(x,y), y + v(x,y))| d\Omega \right\} \quad (32)$$

Figure 20 shows a H&E image and its equivalent vector field.

Figure 20: Working of TVL1 Optical Flow method, a) Given Image; b) Vector Field

4.2. Dataset

H&E and mIF TMA sections for various cancer types were provided by Dr Joe Yeong's team from the Institute of Molecular and Cellular Laboratory (IMCB) at A*Star in large tiff. (Tagged Image File Format) image files. Due to the file size and lack of coordinates regarding the positions of individual sections, each TMA section was manually cropped out to the closest circumferential value using an open-source image editing software called GIMP (GNU Image Manipulation Program) developed by Spencer Kimball and Peter Mattis. Figure 4 and Figure 5 are some samples of the given dataset. Table 5 shows the type and number of TMA sections used. In the last column, the number of omitted sections is mentioned due to the following reasons:

1. Staining issues in H&E and/or mIF images leading to damaged images: Improper staining either on the IF and H&E side for which indications were given by the clinical co-adviser to decide which images are to be rejected, an example of such a discarded sample is Figure 21.

Figure 21: Discarded Image Example (Improper Staining)

2. DAPI regions not bright enough to be picked up by image registration: Corresponding IF regions, especially DAPI regions were too dark and will work poorly with registration.
3. Poor registration output: Blurred or black regions in H&E images which rendered the images unusable, even if the corresponding IF images were clear. an example of such a discarded sample is Figure 22.

Figure 22: Discarded Image Example (Poor Registration)

All these selections were done using visual verification as these samples according to the clinical co-advisor were not beneficial to the study, hence there would not be any biasness in the study.

Table 5 Dataset Details

Type of Cancer	Total Number of Sections	Discarded Sections
TNBC	18	13
Colorectal	44	14
Lung	75	18
Total	137	45

4.3. Proposed Methodology

Image registration is a part of the overall pre-processing pipeline developed using python. Figure 24 shows the proposed workflow for image registration.

1. K-means clustering of H&E images: This method of clustering is used to effectively separate different colours of the images and hence extract the blue nuclei regions of the given image without much input. However, while k-means clustering divides the image into different clusters, as seen in Figure 11, the pre-processing algorithm does not know which cluster has to be taken. Hence using the cluster centre RGB values, the Euclidean distance is calculated between all possible shades of blue and dark purple and the cluster centre

having the least distance is taken to be the required image. Some of these shades with their RGB values are shown in Figure 23.

Figure 23: Shades of Blue with RGB values

The following libraries with their functions and parts of python code are shown:

- a. OpenCV: Open-source image processing software where several pre-processing and k-means clustering is performed. K-means clustering is performed for certain iterations till it reaches a certain epsilon value which refers to the closeness of the calculated centroid to the actual centroid of the given data.
- b. Math: This built-in library was used to calculate the Euclidean distances between the given centroid and the reference shades.
2. Creation of binary masks: The blue nuclei region from the H&E image and the DAPI channel mIF image are first converted to grayscale format followed by the application of Otsu thresholding to create binary masks using OpenCV. Finally, a small 5x5 gaussian blur kernel is applied to remove any holes and bind close-by region to improve image registration.
3. TVL¹ Optical Flow Based Image Registration: The binary masks from the previous step are registered using the following libraries:
 - a. Scikit-image: Another image-processing library in which the optical-flow based registration function is possible. The H&E image mask is taken as the reference image while the DAPI channel mask is taken as the moving image. The required vectors v and u for warping are the outputs of the function. These vectors are then used to warp the main DAPI channel image and all the other channel images, by dividing the RGB color space into three separate images to save image intensity data.
 - b. Matplotlib: This library was used for storing the images for post-processing and statistical processes. While OpenCV and scikit-image have image storing functions, matplotlib required no conversions of the image and was relatively faster compared to these functions.
 - c. OpenCV: The merger of the three separate RGB channels to one image was possible due to this library.

The pre-processing script is available in appendix 1 for reference.

Figure 24: Image Registration Flow

4.4. Results

While most images were registered well for analysis, to really analyse the degree of registration, a simple approach was opted. One of the masks was coloured in red which bears the RGB value of (255,0,0) while the other was coloured in blue which bears the RGB value of (0,0,255). Hence, when an overlap of both coloured masks was performed, the overlapped regions will bear the RGB value of (255,0,255) which is magenta. Therefore, a normalized count of magenta pixels and the ratio between the magenta pixels and the blue pixels gives the degree of overlap. Furthermore, a simple visual verification shows larger number of magenta pixels in the overlapped image after registration. would provide an insight on the degree of overlap, leading to the possibility of creating thresholds with which only the best images can be taken for further analysis.

Figure 25 shows an example of the proposed image registration in work, while Figure 26 shows an example of overlapping, before and after registration.

Table 6 and Table 7 provide a snippet of the calculated red, blue and magenta values, and their normalized values, calculated in Microsoft Excel.

a.

b.

c.

d.

e.

f.

Figure 25: Proposed Image Registration Sample, a) mIF image [top layer]; b) H&E image [bottom layer]; c) Before Registration; d) After Registration; e) Before Registration [3:1 Scale]; f) After Registration [3:1 Scale]

a.

b.

Figure 26: Overlap of Masks Before and After Registration, a) Before Registration; b) After Registration

Table 6: Sample Values of Pixels for Overlap Analysis: Raw Values

Before warping (Raw)			After warping (Raw)		
Magenta	Red	Blue	Magenta	Red	Blue
159268	1234399	580539	1213	478	2387
111314	1127792	578958	1481	479	2877
78458	483873	333508	547	264	1054
302509	1572516	787659	1303	599	2584

Table 7: Sample Values of Pixels for Overlap Analysis: Normalized Values

Before warping (Normalized between -1 to 1)			After warping (Normalized between -1 to 1)		
Magenta	Red	Blue	Magenta	Red	Blue
-0.36333	-0.00527	-0.59193	0.362213	-0.3779	0.282671
-0.63207	-0.17401	-0.59449	1.175399	-0.37498	1.010906
-0.81621	-1.1932	-0.99152	-1.65862	-1.00317	-1.69843
0.43943	0.529892	-0.25689	0.635298	-0.02437	0.575451

Table 8 summarizes the metrics, calculated using the whole dataset consisting of 91 images. In short, there is indeed an improvement with the degree of overlap between the images before and after registration. Figure 27 are box plots of each evaluation metric which can provide a comprehensive analysis on the data derived, showing the mean values and the spread of scores before and after alignment.

Figure 27: Boxplot of Similarity Measures

Table 8: Mean of Similarity Measures

Method	Before Registration (mean)	After Registration (mean)
SSIM	0.591	0.778
PSNR	7.319	13.255
Ratio	0.192	0.330

Chapter Five

CD3 Region Prediction on H&E Images

5.1. Generative Adversarial Networks

A Generative Adversarial Network (GAN) is a special type of network as it consists of two distinct neural networks working simultaneously against each other. These components are:

1. Generator: In the case of image-to-image translation, this network works based on the given image being converted to another image via a function as seen in equation (33). Hence the objective of this network is to optimize the function to generate a specific set of transformed images.

$$G: x \rightarrow y \quad (33)$$

2. Discriminator: The function of this network is to determine the probability of a given image to be either real or generated from the generator.

Therefore, the combined objective of the network is for the generator to properly create images in a way that the discriminator is unable to distinguish if the given images are real or are from the generator. Hence two losses are closely monitored, the loss of the generator and that of the discriminator. However, it should be noted that neither of the losses should be very minimal, as in no one network should do better than the other, e.g., the generator could supposedly prepare garbage images and still have a lower loss, but the discriminator easily dismisses the generated images as being fake, thereby increasing its loss [94,95]. Equation (34) gives the objective function V of a GAN.

$$\min_G \left(\max_D V(D, G) \right) = E_{x \sim p_{\text{data}}(x)} [\log D(x)] + E_{z \sim p_z(z)} [\log(1 - D(G(z)))] \quad (34)$$

Where D and G are the discriminator and generator functions, respectively. The given point in a dataset x and its probability distribution $p_{\text{data}}(x)$ is converted to a point in a latent space z with probability distribution $p_z(z)$. The loss function in use is the binary-cross entropy function and the tanh function is used as the activation function.

A conditional GAN is a variation of the original GAN algorithm, but with class information added. Hence, the discriminator also must process class information with the image resulting in the use of an additional loss function for the classification errors. A variation of this network called as Pix2Pix is used frequently for image generation tasks as it replaces the usual GAN generator with an encoder-decoder structure and replaces class information with the image from the other domain, which is the output

image [95,96]. Hence the losses are more meaningful to generate transformed images. A point to note is that batch normalization is used in this network. Equation (35) shows the overall loss function while equations (36) and (37) show the conditional and $\mathcal{L}_1$ generator loss, respectively.

$$G^* = \arg \min_G \max_D \mathcal{L}_{\text{cGAN}}(G, D) + \lambda \mathcal{L}_{L1}(G) \quad (35)$$

$$\mathcal{L}_{\text{cGAN}}(G, D) = E_y[\log D(y)] + E_{x,z}[\log(1 - D(G(x, z)))] \quad (36)$$

$$\mathcal{L}_{L1}(G) = E_{x,y,z}[\|y - G(x, z)\|_1] \quad (37)$$

Where λ is used to ensure minimal chances of overfitting by multiplying big numbers with the $\mathcal{L}_1$ loss to dramatically increase loss and make the network work more for loss minimization.

A popular encoder-decoder structure used for Pix2Pix GANs are U-Nets, as shown in Figure 28. These networks are identified by its unique U shape, starting with encoding path from top to bottom, followed by a decoding path from bottom to top.

Figure 28: U-net Generator Structure [97]

Another type of GAN which is of particular interest is cycleGAN, in which unpaired image translation is a possibility, which means that image pairs without being accurately registered can also be used for training. This is achieved with the help of two additional loss functions, a cycle consistency loss function, in which an image generated from a generator is run into another generator to obtain the initial image and hence the loss is calculated based on the error value obtained. The combined identity loss is the addition of computed identity losses from both domains. A point to note is that unlike batch normalization in Pix2Pix, cycleGAN uses instance normalization [98,99]. Equation (38) shows the combined loss, in which the first two

terms are GAN based losses similar to equation (35). Equation (39) shows the cycle consistency loss.

$$G^* = \arg \min_G \max_D \mathcal{L}_{cGAN}(G, D_X, X, Y) + \mathcal{L}_{cGAN}(G, D_Y, Y, X) + \lambda \mathcal{L}_{cyc}(G, F) \quad (38)$$

$$\mathcal{L}_{cyc}(G, F) = E_{x \sim p_{data}(x)} [\|F(G(x)) - x\|_1] + E_{y \sim p_{data}(y)} [\|G(F(y)) - y\|_1] \quad (39)$$

Where X is the given image and Y is the reference image. The functions G and D_x form the first generator-discriminator pair, converting the given image to the reference image, while the functions F and D_y form the second generator-discriminator pair, converting the reference image back to the initial given image.

All algorithms here use the Adam optimization algorithm to update weights.

5.2. Proposed Methodology

A combination of the previous chapter for pre-processing followed by the creation of ground truth images and finally training of networks is the overall process, as shown in Figure 29 and Figure 30.

1. Pre-processing: Optical flow-based Image Registration as shown in the previous chapter was used.
2. Creation of Ground Truth Images: For a dataset (same dataset used for image registration) of 15000 images to be compatible to Pix2Pix and CycleGAN, samples like that in Figure 31 are prepared. CD3 regions procured from registered mIF images are superimposed on its corresponding H&E image.

As CD3 regions seem to have no structural resemblance in H&E images, a blurred version of the regions is used while retaining intensity using features like Gaussian blur using OpenCV to ensure repeatability. Usage of single channel images solely as the ground truth especially for cycleGAN was not useful since the cycle consistency loss would be very high due to the lack of data about other regions. Hence a possible solution would be to add all channels possible, but that would lead to another post-processing step of removing the required regions, possibly by k-means and clusters and apart from DAPI regions, there is less structural resemblance of other mIF channel regions with the H&E. Hence the superimposed image with an approximate blur was an ideal choice. Two levels of blurring, based on kernel size were used, one with size 101 and one with 255 and with appropriate adjustments in brightness. Superimposition with adjustment in brightness was done with the help of OpenCV's "addWeighted" function. Most of the raw images, had an approximate length and breadth of 3000 – 3500 pixels.

For effective image analysis, the dataset was divided into tiles of size 256x256. Using OpenCV, images were resized to 3328x3328 pixels with respect to the

average dimensions as there would be no distortion in the size of tiles, especially towards the edges, resulting in close to 16000 tiles.

Figure 29: Process Workflow

Figure 30: Deep Learning Process Flow

Figure 31: Ground Truth Image Examples, a) Low Brightness, Low Blurring; b) Low Brightness, High Blurring; c) High Brightness, Low Blurring; d) High Brightness, High Blurring

3. Training and Validation: The following libraries were used for this purpose:

- a. Sklearn: To split the dataset for training, sklearn's test-train-split function was used with the split being 80% for training and rest for validation.
- b. Tensorflow: The Pix2Pix and cycleGAN network details, with details of each layer written in python using Google's deep learning framework called as Tensorflow (version 2.4.1) are given in Appendix 2. This library helps in the construction of the networks and readymade loss functions and similarity measures to understand the progress of the network. Furthermore, simple data augmentation processes like flipping and rotating images and even subjecting images to random brightness and contrast could be done while learning using this library.

Tensorflow is a python library which can fully take advantage of extra computational power using Graphics Processing Units (GPUs). Using capable GPUs, from the likes of

NVIDIA and its special research based Quadro GPUs, larger batches of data could be trained in one loop with speeds unparalleled compared to Central Processing Units (CPUs) speeds. In the case of Pix2Pix GANs, batch size of 100 was chosen which took only 90 seconds per loop. However, due to the complexity of cycleGAN, only batches of up to 25 were possible to be processed per loop which took close to 700 seconds.

During training, no mention is made to the network by any means, say vectors to indicate the type of cancer, hence the network seems to work well on at least three cancer types, that of triple negative breast cancer, lung cancer and colorectal cancer. Study of literature and comments from the clinical co-advisor concluded that there would not be a discernable anomaly if the model were to work with one or multiple cancer types. Furthermore, the reason why multiple cancer types were used in the study is to increase the total number of samples so that the GAN based models could work better.

5.3. Results

The CD3 region prediction task can be assumed as an image segmentation task in which the usage of the accuracy metric was not advisable as the scores would unbearably increase or decrease due to blank areas and other negative areas. Additionally, varying intensities of the CD3 regions also exist, unlike usual image segmentation tasks whose reference images contain masks comprising of different colours but with specific intensities. Hence metrics like SSIM and PSNR are used which are better indicators of the algorithm's working.

Keeping in mind that neither the generator nor discriminator works better than the other, the individual losses were closely monitored. It was also noticed that a perfect alignment was not needed even in the case of Pix2Pix GAN algorithm and hence the main operation taking place was the process of style transfer from the target image on to the input image. Table 9 shows the SSIM and PSNR values of the various models used on different datasets. High values of SSIM as it is closer to 1 and PSNR values being as high as 45 means that all the models do a formidable job in predicting approximated CD3 regions. A point to note is that all comparisons are done with each model running the same number of loops. Due to cycleGAN having almost double number of steps compared to a regular GAN model, the time taken per loop is close to ten times as much as that of the regular GAN model.

Table 9: Statistical Data from Algorithms

Dataset		GAN		CycleGAN	
Brightness	Blur	PSNR	SSIM	PSNR	SSIM
Low	Low	44.07416	0.985277	23.91133	0.800102
High	Low	40.35224	0.97303	25.79459	0.901593
Low	High	48.1779	0.990045	31.76675	0.932537
High	High	39.31528	0.956801	25.30454	0.890048

Table 10 shows some CD3 negative regions while Table 11 shows some CD3 positive regions, selected based on the higher values of SSIM. Heatmaps are used for visual comparison between the reference and predicted images. Since CD3 regions are marked in green, the heatmaps show the distribution of a given image in the green channel of RGB. c. d.

Figure 32 is an example of the working of the algorithm on a TMA sample. For zoomed in negative regions as in Table 10, the maximum value observed in the heatmap is lesser than that of Table 11, which contain CD3 positive regions.

a.

b.

c.

d.

Figure 32: Whole TMA Prediction Example; a) Reference Image; b) Predicted Image; c) 50x50 Heat Map of Reference Image; d) 50x50 Heat Map of Predicted Image

Table 10: CD3 Negative Regions Examples

Sn No	Given Image	Reference and Predicted Image Heat Maps	Reference Image	Registered CD3 Regions
1
2

Table 11: CD3 Positive Regions Examples

Sn No	Given Image	Reference and Predicted Image Heat Maps	Reference Image	Threshold CD3 Regions
1
2

Chapter Six

Conclusions

6.1. Contributions

In an effort to battle cancer, diagnostic tools are essential. However, each examination and its subsequent treatment is expensive, leaving some patients to not explore the possibilities of newer treatment techniques like immunotherapy. Hence, studies on medical images using machine learning paradigms prove to be useful as it could not only derive valuable information which was previously not found, but it could also be used to significantly reduce costs of diagnosis as trained algorithms could provide diagnostic information which are usually discovered using sophisticated equipment, something which the contents of this thesis aim to fulfil.

The summary of contributions of this thesis is as follows:

1. Image Registration: Usage of optical flow vectors to automatically and efficiently co-register H&E and mIF images, validated with the increase in the PSNR and SSIM values. The following are the highlights of this study:
 - a. Use of Histopathological Images: While optical flow registration methods are used in other imaging modalities, their use in histopathological images has not yet been fully explored, making this study one of the few to investigate this method.
 - b. Reusability: this technique is compatible with the dataset provided on which not a lot of deep learning studies are done, hence creating sets of images which can be used for other studies as well.
 - c. Automation: The image registration algorithm only needs raw images as input, the whole process takes place with little to no human input.
 - d. Speed: Compared to feature-based registration methods like SURF or SIFT, this intensity-based registration method is significantly faster and yet it provides good results.
2. Deep Learning: Usage of GAN based algorithms to predict CD3 regions using only H&E images, also verified with the help of SSIM and PSNR values. The following are the highlights of this study:
 - a. Use of CycleGAN: While GAN based methods are being explored for histopathological images, the use of cycleGAN is rare due to it being relatively newer compared to other deep learning frameworks, making this amongst the few studies to focus on this direction.

- b. GPU Acceleration: Coupled with power GPU accelerated learning, several datasets and combinations were tested with relative ease and only the best combinations are highlighted in this study.
- c. Reusability: The python scripts are optimized to handle any variation of histopathological images.
- d. Expenses: With a possibility of effectively converting images from one modality to another, this study opens the possibility of conducting mIF based examination on patients with little to no cost, significantly reducing the cost of cancer treatment.

6.2. Limitations and Future Work

While feature-based image registration methods are superior to intensity-based image registration methods, the work of this thesis shows that the latter could also be used which leads to overall faster registration times. However, there would be some inconsistencies in registration due to differences in textures or quality of images, hence a possible threshold based on any one of the similarity measures used can help in the procurement of only the best images for analysis. The only issue here is that a proper threshold value needs to be decided so that not a lot of the dataset is being discarded. Furthermore, image registration procedures happen only in the CPU since most functions and libraries like OpenCV or skimage have not fully migrated to GPU based scripts which could significantly reduce the hours it takes currently to register images and evaluate the quality of registration.

In the field of deep learning, Pix2Pix GANs for the same number of loops seem to function better than cycleGANs, but due to the complexity of cycleGANs and the amount of time it takes per loop, multi-GPU solutions could be used to run cycleGANs for longer durations which may potentially produce better results than Pix2Pix GANs. However, for the python script to support multi-GPU solutions, significant number of changes need to be made. Furthermore, there are several hyperparameters in both networks which can be altered to provide better results. Additionally, the ideal solution would be for the CD3 regions to be detected in H&E images without any blur, or at least with minimal blur. Hence several tests need to be conducted to estimate the right amount of blurring and brightness to produce data closer to what would be found in mIF images.

Usage of pathological images with optical flow and cycleGAN is not fully explored based on literature surveys, hence, to standardize hyperparameters and the type of dataset used, trial and error method would have to be followed.

Finally, this method could be applied to several other regions like CD20 or CD45 regions which have been previously mentioned to also help in cancer diagnosis. Hence, this

CAD method could be potentially used as a preliminary test, whereby if positive regions are shown, only then mIF images are prepared, therefore significantly reducing the cost of cancer treatment especially in third-world countries.

References

- [1] "Definition of Cancer - NCI Dictionary of Cancer Terms - National Cancer Institute" [Online]. Available: <https://www.cancer.gov/publications/dictionaries/cancer-terms/def/cancer>. [Accessed: 12-Apr-2021].
- [2] "What Is Cancer? | Cancer.Net" [Online]. Available: <https://www.cancer.net/navigating-cancer-care/cancer-basics/what-cancer>. [Accessed: 12-Apr-2021].
- [3] Patel, A., 2020, "Benign vs Malignant Tumors," *JAMA Oncol.*, **6**(9), p. 1488.
- [4] Sung, H., Ferlay, J., Siegel, R. L., Laversanne, M., Soerjomataram, I., Jemal, A., and Bray, F., 2021, "Global Cancer Statistics 2020: GLOBOCAN Estimates of Incidence and Mortality Worldwide for 36 Cancers in 185 Countries," *CA. Cancer J. Clin.*
- [5] KOH, P. Y., 2019, "PRINCIPAL CAUSES OF DEATH," p. 1 [Online]. Available: <https://www.moh.gov.sg/resources-statistics/singapore-health-facts/principal-causes-of-death>. [Accessed: 26-Apr-2021].
- [6] Mehta, S. R., Suhag, V., Semwal, M., and Sharma, N., 2010, "Radiotherapy: Basic Concepts and Recent Advances," *Med. J. Armed Forces India*, **66**(2), pp. 158–162.
- [7] Huang, C. Y., Ju, D. T., Chang, C. F., Muralidhar Reddy, P., and Velmurugan, B. K., 2017, "A Review on the Effects of Current Chemotherapy Drugs and Natural Agents in Treating Non-Small Cell Lung Cancer," *Biomed.*, **7**(4), pp. 12–23.
- [8] Joo, W. D., Visintin, I., and Mor, G., 2013, "Targeted Cancer Therapy - Are the Days of Systemic Chemotherapy Numbered?," *Maturitas*, **76**(4), pp. 308–314.
- [9] Puhalla, S., Bhattacharya, S., and Davidson, N. E., 2012, "Hormonal Therapy in Breast Cancer: A Model Disease for the Personalization of Cancer Care," *Mol. Oncol.*, **6**(2), pp. 222–236.
- [10] Emens, L. A., Ascierto, P. A., Darcy, P. K., Demaria, S., Eggermont, A. M. M., Redmond, W. L., Seliger, B., and Marincola, F. M., 2017, "Cancer Immunotherapy: Opportunities and Challenges in the Rapidly Evolving Clinical Landscape," *Eur. J. Cancer*, **81**, pp. 116–129.
- [11] Miller, K. D., Nogueira, L., Mariotto, A. B., Rowland, J. H., Yabroff, K. R., Alfano, C. M., Jemal, A., Kramer, J. L., and Siegel, R. L., 2019, "Cancer Treatment and Survivorship Statistics, 2019," *CA. Cancer J. Clin.*, **69**(5), pp. 363–385.
- [12] Baghban, R., Roshangar, L., Jahanban-Esfahlan, R., Seidi, K., Ebrahimi-Kalan, A., Jaymand, M., Kolahian, S., Javaheri, T., and Zare, P., "Tumor Microenvironment Complexity and Therapeutic Implications at a Glance."
- [13] Hui, L., and Chen, Y., 2015, "Tumor Microenvironment: Sanctuary of the Devil," *Cancer Lett.*, **368**(1), pp. 7–13.

- [14] Yue, B., 2014, "Biology of the Extracellular Matrix: An Overview," *J. Glaucoma*, **23**(8), pp. S20–S23.
- [15] Gál, P., Varinská, L., Fáber, L., Novák, Š., Szabo, P., Mitrengová, P., Mirossay, A., Mučaji, P., and Smetana, K., 2017, "How Signaling Molecules Regulate Tumor Microenvironment: Parallels to Wound Repair," *Molecules*, **22**(11).
- [16] Nagl, L., Horvath, L., Pircher, A., and Wolf, D., 2020, "Tumor Endothelial Cells (TECs) as Potential Immune Directors of the Tumor Microenvironment – New Findings and Future Perspectives," *Front. Cell Dev. Biol.*, **8**, p. 766.
- [17] Meng, M. Bin, Zaorsky, N. G., Deng, L., Wang, H. H., Chao, J., Zhao, L. J., Yuan, Z. Y., and Ping, W., 2015, "Pericytes: A Double-Edged Sword in Cancer Therapy," *Futur. Oncol.*, **11**(1), pp. 169–179.
- [18] Lin, Y., Xu, J., and Lan, H., "Tumor-Associated Macrophages in Tumor Metastasis: Biological Roles and Clinical Therapeutic Applications."
- [19] Takahama, H., Sasaki, F., and Kinoshita, T., 1992, "Structural and Endocytotic Differences of Fibroblasts and Macrophages in the Tail Fin of Amphibian Larvae during Metamorphosis," *Arch. Histol. Cytol.*, **55**(4), pp. 437–448.
- [20] Sahai, E., Astsaturov, I., Cukierman, E., DeNardo, D. G., Egeblad, M., Evans, R. M., Fearon, D., Greten, F. R., Hingorani, S. R., Hunter, T., Hynes, R. O., Jain, R. K., Janowitz, T., Jorgensen, C., Kimmelman, A. C., Kolonin, M. G., Maki, R. G., Scott Powers, R., Puré, E., Ramirez, D. C., Scherz-Shouval, R., Sherman, M. H., Stewart, S., Tlsty, T. D., Tuveson, D. A., Watt, F. M., Weaver, V., Weeraratna, A. T., and Werb, Z., "A Framework for Advancing Our Understanding of Cancer-Associated Fibroblasts."
- [21] Anderson, N. M., and Simon, M. C., 2020, "The Tumor Microenvironment," *Curr. Biol.*, **30**(16), pp. R921–R925.
- [22] Wong, R. S. Y., 2011, "Apoptosis in Cancer: From Pathogenesis to Treatment," *J. Exp. Clin. Cancer Res.*, **30**(1), p. 87.
- [23] Fernandez-Garcia, D., Hills, A., Page, K., Hastings, R. K., Toghiani, B., Goddard, K. S., Ion, C., Ogle, O., Boydell, A. R., Gleason, K., Rutherford, M., Lim, A., Guttery, D. S., Coombes, R. C., and Shaw, J. A., 2019, "Plasma Cell-Free DNA (CfDNA) as a Predictive and Prognostic Marker in Patients with Metastatic Breast Cancer," *Breast Cancer Res.*, **21**(1), p. 149.
- [24] "Helper and Cytotoxic T Cells | British Society for Immunology" [Online]. Available: <https://www.immunology.org/public-information/bitesized-immunology/cells/helper-and-cytotoxic-t-cells>. [Accessed: 18-Apr-2021].
- [25] 2004, "CLUSTER OF DIFFERENTIATION (CD) ANTIGENS," *Immunology Guidebook*, Elsevier, pp. 47–124.
- [26] Lloyd, E. C., Gandhi, T. N., and Petty, L. A., 2021, "Monoclonal Antibodies for COVID-19," *JAMA - J. Am. Med. Assoc.*, **325**(10), p. 1015.
- [27] Singh, H., Charaya, N., Poonia, M., Kaur Sidhu, S., and Singh Sihmar, S., 2016,

Biopsy-A Vision of Life.

- [28] VanderLaan, P. A., 2016, "Fine-Needle Aspiration and Core Needle Biopsy: An Update on 2 Common Minimally Invasive Tissue Sampling Modalities," *Cancer Cytopathol.*, **124**(12), pp. 862–870.
- [29] "Endoscopic Biopsy | Stanford Health Care" [Online]. Available: <https://stanfordhealthcare.org/medical-tests/b/biopsy/types/endoscopic-biopsy.html>. [Accessed: 27-Apr-2021].
- [30] Nischal, U., Nischal, K., and Khopkar, U., 2008, "Techniques of Skin Biopsy and Practical Considerations," *J. Cutan. Aesthet. Surg.*, **1**(2), p. 107.
- [31] Ignatiadis, M., Sledge, G. W., and Jeffrey, S. S., 2021, "Liquid Biopsy Enters the Clinic — Implementation Issues and Future Challenges," *Nat. Rev. Clin. Oncol.*, **18**(5), pp. 297–312.
- [32] Sikandar, A., 2018, "Histopathology: An Old Yet Important Technique in Modern Science," *Histopathology - An Update*, InTech.
- [33] He, L., Long, L. R., Antani, S., and Thoma, G. R., 2012, "Histology Image Analysis for Carcinoma Detection and Grading," *Comput. Methods Programs Biomed.*, **107**(3), pp. 538–556.
- [34] Mukhopadhyay, S., Feldman, M. D., Abels, E., Ashfaq, R., Beltaifa, S., Cacciabeve, N. G., Cathro, H. P., Cheng, L., Cooper, K., Dickey, G. E., Gill, R. M., Heaton, R. P., Kerstens, R., Lindberg, G. M., Malhotra, R. K., Mandell, J. W., Manlucu, E. D., Mills, A. M., Mills, S. E., Moskaluk, C. A., Nelis, M., Patil, D. T., Przybycin, C. G., Reynolds, J. P., Rubin, B. P., Saboorian, M. H., Salicru, M., Samols, M. A., Sturgis, C. D., Turner, K. O., Wick, M. R., Yoon, J. Y., Zhao, P., and Taylor, C. R., 2018, "Whole Slide Imaging Versus Microscopy for Primary Diagnosis in Surgical Pathology," *Am. J. Surg. Pathol.*, **42**(1), pp. 39–52.
- [35] "ImmunoAtlas" [Online]. Available: <https://immunoatlas.org/JEDI/201207-1/JEDI20033/>. [Accessed: 28-Apr-2021].
- [36] Barrette, K., Van Den Oord, J. J., and Garmyn, M., 2014, "Tissue Microarray," *J. Invest. Dermatol.*, **134**, pp. 1–4.
- [37] Khouja, M. H., Baekelandt, M., Sarab, A., Nesland, J. M., and Holm, R., 2010, "Limitations of Tissue Microarrays Compared with Whole Tissue Sections in Survival Analysis," *Oncol. Lett.*, **1**(5), pp. 827–831.
- [38] Fischer, A. H., Jacobson, K. A., Rose, J., and Zeller, R., 2008, "Hematoxylin and Eosin Staining of Tissue and Cell Sections," *Cold Spring Harb. Protoc.*, **3**(5).
- [39] Kaliyappan, K., Palanisamy, M., Duraiyan, J., and Govindarajan, R., 2012, "Applications of Immunohistochemistry," *J. Pharm. Bioallied Sci.*, **4**(6), p. 307.
- [40] Yeong, J., Thike, A. A., Lim, J. C. T., Lee, B., Li, H., Wong, S. C., Hue, S. S. S., Tan, P. H., and Iqbal, J., 2017, "Higher Densities of Foxp3+ Regulatory T Cells Are Associated with Better Prognosis in Triple-Negative Breast Cancer," *Breast Cancer Res. Treat.*, **163**(1), pp. 21–35.

- [41] Tan, W. C. C., Nerurkar, S. N., Cai, H. Y., Ng, H. H. M., Wu, D., Wee, Y. T. F., Lim, J. C. T., Yeong, J., and Lim, T. K. H., 2020, "Overview of Multiplex Immunohistochemistry/Immunofluorescence Techniques in the Era of Cancer Immunotherapy," *Cancer Commun.*, **40**(4), pp. 135–153.
- [42] "Average Cost of Cancer Treatment in Singapore | ValueChampion Singapore" [Online]. Available: <https://www.valuechampion.sg/average-cost-cancer-treatment-singapore>. [Accessed: 30-Apr-2021].
- [43] "Reality Check: Cost of Battling Cancer in Singapore | DBS Singapore" [Online]. Available: <https://www.dbs.com.sg/personal/articles/nav/protection/cost-of-battling-cancer-in-singapore#>. [Accessed: 30-Apr-2021].
- [44] Cooper, L., Sertel, O., Kong, J., Lozanski, G., Huang, K., and Gurcan, M., 2009, "Feature-Based Registration of Histopathology Images with Different Stains: An Application for Computerized Follicular Lymphoma Prognosis," *Comput. Methods Programs Biomed.*, **96**(3), pp. 182–192.
- [45] Keikhosravi, A., Li, B., Liu, Y., and Eliceiri, K. W., 2020, "Intensity-Based Registration of Bright-Field and Second-Harmonic Generation Images of Histopathology Tissue Sections," *Biomed. Opt. Express*, **11**(1), p. 160.
- [46] Peñaranda, F., Naranjo, V., Verdú-Monedero, R., Lloyd, G. R., Nallala, J., and Stone, N., 2017, "Multimodal Registration of Optical Microscopic and Infrared Spectroscopic Images from Different Tissue Sections: An Application to Colon Cancer," *Digit. Signal Process. A Rev. J.*, **68**, pp. 1–15.
- [47] Jiang, J., Larson, N. B., Prodduturi, N., Flotte, T. J., and Hart, S. N., 2019, "Robust Hierarchical Density Estimation and Regression for Re-Stained Histological Whole Slide Image Co-Registration," *PLoS One*, **14**(7), p. e0220074.
- [48] Wang, C.-W., and Chen, H.-C., 2013, "Improved Image Alignment Method in Application to X-Ray Images and Biological Images," *Bioinformatics*, **29**(15), pp. 1879–1887.
- [49] Hermann, J., Brehmer, K., Jankowski, V., Lellig, M., Hohl, M., Mahfoud, F., Speer, T., Schunk, S. J., Tschernig, T., Thiele, H., and Jankowski, J., 2021, "Registration of Image Modalities for Analyses of Tissue Samples Using 3D Image Modelling," *Proteomics - Clin. Appl.*, **15**(1).
- [50] Lotz, J., Olesch, J., Muller, B., Polzin, T., Galuschka, P., Lotz, J. M., Heldmann, S., Laue, H., Gonzalez-Vallinas, M., Warth, A., Lahrmann, B., Grabe, N., Sedlacek, O., Breuhahn, K., and Modersitzki, J., 2016, "Patch-Based Nonlinear Image Registration for Gigapixel Whole Slide Images," *IEEE Trans. Biomed. Eng.*, **63**(9), pp. 1812–1819.
- [51] Jackson, C. R., Sriharan, A., and Vaickus, L. J., 2020, "A Machine Learning Algorithm for Simulating Immunohistochemistry: Development of SOX10 Virtual IHC and Evaluation on Primarily Melanocytic Neoplasms," *Mod. Pathol.*, **33**(9), pp. 1638–1648.
- [52] Burlingame, E. A., McDonnell, M., Schau, G. F., Thibault, G., Lanciault, C.,

- Morgan, T., Johnson, B. E., Corless, C., Gray, J. W., and Chang, Y. H., 2020, "SHIFT: Speedy Histological-to-Immunofluorescent Translation of a Tumor Signature Enabled by Deep Learning," *Sci. Rep.*, **10**(1), pp. 1–14.
- [53] Chang, Y. H., Burlingame, E. A., Gray, J. W., and Margolin, A. A., 2018, "SHIFT: Speedy Histopathological-to-Immunofluorescent Translation of Whole Slide Images Using Conditional Generative Adversarial Networks."
- [54] Litjens, G., Kooi, T., Bejnordi, B. E., Setio, A. A. A., Ciompi, F., Ghafoorian, M., van der Laak, J. A. W. M., van Ginneken, B., and Sánchez, C. I., 2017, "A Survey on Deep Learning in Medical Image Analysis," *Med. Image Anal.*, **42**, pp. 60–88.
- [55] Pereira, S., Pinto, A., Alves, V., and Silva, C. A., 2016, "Brain Tumor Segmentation Using Convolutional Neural Networks in MRI Images," *IEEE Trans. Med. Imaging*, **35**(5), pp. 1240–1251.
- [56] Yoo, Y., Tang, L. W., Brosch, T., Li, D. K. B., Metz, L., Traboulsee, A., and Tam, R., 2016, "Deep Learning of Brain Lesion Patterns for Predicting Future Disease Activity in Patients with Early Symptoms of Multiple Sclerosis," *Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics)*, **10008 LNCS**, pp. 86–94.
- [57] Gao, X., Lin, S., and Wong, T. Y., 2015, "Automatic Feature Learning to Grade Nuclear Cataracts Based on Deep Learning," *IEEE Trans. Biomed. Eng.*, **62**(11), pp. 2693–2701.
- [58] T, S., SM, W., WD, V., U, S.-E., and G, L., 2015, "Predicting Semantic Descriptions from Medical Images with Convolutional Neural Networks," *Inf. Process. Med. Imaging*, **24**, pp. 437–448.
- [59] Lee, J.-G., Jun, S., Cho, Y.-W., Lee, H., Kim, G. B., Seo, J. B., and Kim, N., 2017, "Deep Learning in Medical Imaging: General Overview," *Korean J. Radiol.*, **18**(4), p. 570.
- [60] Chaunzwa, T. L., Hosny, A., Xu, Y., Shafer, A., Diao, N., Lanuti, M., Christiani, D. C., Mak, R. H., and Aerts, H. J. W. L., 2021, "Deep Learning Classification of Lung Cancer Histology Using CT Images," *Sci. Reports 2021 111*, **11**(1), pp. 1–12.
- [61] Wang, J., Ding, H., Bidgoli, F. A., Zhou, B., Iribarren, C., Molloy, S., Baldi, P., Fellow, and IEEE, 2017, "Detecting Cardiovascular Disease from Mammograms with Deep Learning," *IEEE Trans. Med. Imaging*, **36**(5), p. 1172.
- [62] Pogorelov, K., Ostroukhova, O., Jeppsson, M., Espeland, H., Griwodz, C., De Lange, T., Johansen, D., Riegler, M., and Halvorsen, P., 2018, "Deep Learning and Hand-Crafted Feature Based Approaches for Polyp Detection in Medical Videos," *Proc. - IEEE Symp. Comput. Med. Syst.*, **2018-June**, pp. 381–386.
- [63] Lahiani, A., Klaiman, E., and Grimm, O., 2018, "Enabling Histopathological Annotations on Immunofluorescent Images through Virtualization of Hematoxylin and Eosin," *J. Pathol. Inform.*
- [64] Selvaraju, R. R., Cogswell, M., Das, A., Vedantam, R., Parikh, D., and Batra, D.,

- 2017, "Grad-CAM: Visual Explanations from Deep Networks via Gradient-Based Localization," *Proc. IEEE Int. Conf. Comput. Vis.*, **2017-October**, pp. 618–626.
- [65] Panwar, H., Gupta, P. K., Khubeb Siddiqui, M., Morales-Menendez, R., Bhardwaj, P., and Singh, V., 2020, "A Deep Learning and Grad-CAM Based Color Visualization Approach for Fast Detection of COVID-19 Cases Using Chest X-Ray and CT-Scan Images," *Chaos, Solitons and Fractals*, **140**, p. 110190.
 - [66] Fernández, R., Rosado, P., Vegas, E., and Reverter, F., 2021, "Medical Image Editing in the Latent Space of Generative Adversarial Networks," *Intell. Med.*, **5**, p. 100040.
 - [67] Wang, D., Khosla, A., Gargeya, R., Irshad, H., and Beck, A. H., 2016, "Deep Learning for Identifying Metastatic Breast Cancer."
 - [68] Choy, G., Khalilzadeh, O., Michalski, M., Do, S., Samir, A. E., Pianykh, O. S., Geis, J. R., Pandharipande, P. V., Brink, J. A., and Dreyer, K. J., 2018, "Current Applications and Future Impact of Machine Learning in Radiology," *Radiology*, **288**(2), pp. 318–328.
 - [69] Nie, D., Trullo, R., Lian, J., Petitjean, C., Ruan, S., Wang, Q., and Shen, D., 2017, "Medical Image Synthesis with Context-Aware Generative Adversarial Networks," *Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, Springer Verlag, pp. 417–425.
 - [70] Nie, D., Trullo, R., Lian, J., Wang, L., Petitjean, C., Ruan, S., Wang, Q., and Shen, D., 2018, "Medical Image Synthesis with Deep Convolutional Adversarial Networks," *IEEE Trans. Biomed. Eng.*, **65**(12), pp. 2720–2730.
 - [71] Chertsias, A., Joyce, T., Dharmakumar, R., and Tsiftaris, S. A., 2017, "Adversarial Image Synthesis for Unpaired Multi-Modal Cardiac Data," *Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, Springer Verlag, pp. 3–13.
 - [72] Costa, P., Galdran, A., Meyer, M. I., Niemeijer, M., Abràmoff, M., Mendonça, A. M., and Campilho, A., 2018, "End-to-End Adversarial Retinal Image Synthesis," *IEEE Trans. Med. Imaging*, **37**(3), pp. 781–791.
 - [73] Khosravi, P., Lysandrou, M., Eljalby, M., Li, Q., Kazemi, E., Zisimopoulos, P., Sigaras, A., Brendel, M., Barnes, J., Ricketts, C., Meleshko, D., Yat, A., McClure, T. D., Robinson, B. D., Sboner, A., Elemento, O., Chughtai, B., and Hajirasouliha, I., 2021, "A Deep Learning Approach to Diagnostic Classification of Prostate Cancer Using Pathology–Radiology Fusion," *J. Magn. Reson. Imaging*, **54**(2).
 - [74] Quiros, A. C., Murray-Smith, R., and Yuan, K., 2019, "PathologyGAN: Learning Deep Representations of Cancer Tissue," *arXiv*, **2021**, pp. 4–5.
 - [75] Levine, A. B., Peng, J., Farnell, D., Nurse, M., Wang, Y., Naso, J. R., Ren, H., Farahani, H., Chen, C., Chiu, D., Talhouk, A., Sheffield, B., Riaz, M., Ip, P. P., Parra-Herran, C., Mills, A., Singh, N., Tessier-Cloutier, B., Salisbury, T., Lee, J., Salcudean, T., Jones, S. J., Huntsman, D. G., Gilks, C. B., Yip, S., and Bashashati,

- A., 2020, "Synthesis of Diagnostic Quality Cancer Pathology Images by Generative Adversarial Networks," *J. Pathol.*, **252**(2), pp. 178–188.
- [76] Rana, A., Yauney, G., Lowe, A., and Shah, P., 2018, "Computational Histological Staining and Destaining of Prostate Core Biopsy RGB Images with Generative Adversarial Neural Networks."
 - [77] Xu, Z., Moro, C. F., Bozóky, B., and Zhang, Q., 2019, "GAN-Based Virtual Re-Staining: A Promising Solution for Whole Slide Image Analysis," *arXiv*.
 - [78] Periaswamy, S., and Farid, H., 2006, "Medical Image Registration with Partial Data," *Med. Image Anal.*, **10**(3 SPEC. ISS.), pp. 452–464.
 - [79] Lukashevich, P. V., Zalesky, B. A., and Ablameyko, S. V., 2011, "Medical Image Registration Based on SURF Detector," *Pattern Recognit. Image Anal.*, **21**(3), pp. 519–521.
 - [80] Vibhakar, A., Tiwari, M., and Singh, J., 2012, "Performance Analysis for MRI Denoising Using Intensity Averaging Gaussian Blur Concept and Its Comparison with Wavelet Transform Method," *Int. J. Comput. Appl.*, **975**, p. 8887.
 - [81] Otsu, N., 1979, "THRESHOLD SELECTION METHOD FROM GRAY-LEVEL HISTOGRAMS.," *IEEE Trans Syst Man Cybern*, **SMC-9**(1), pp. 62–66.
 - [82] Sankur, B., 2004, "Survey over Image Thresholding Techniques and Quantitative Performance Evaluation," *J. Electron. Imaging*, **13**(1), p. 146.
 - [83] Hartigan, J. A., and Wong, M. A., 1979, "Algorithm AS 136: A K-Means Clustering Algorithm," *Appl. Stat.*, **28**(1), p. 100.
 - [84] Yamashita, R., Nishio, M., Do, R. K. G., and Togashi, K., 2018, "Convolutional Neural Networks: An Overview and Application in Radiology," *Insights Imaging*, **9**(4), pp. 611–629.
 - [85] "Cross-Entropy Loss Function. A Loss Function Used in Most... | by Kiprono Elijah Koech | Towards Data Science" [Online]. Available: <https://towardsdatascience.com/cross-entropy-loss-function-f38c4ec8643e>. [Accessed: 06-May-2021].
 - [86] Kingma, D. P., and Ba, J. L., 2015, "Adam: A Method for Stochastic Optimization," *3rd International Conference on Learning Representations, ICLR 2015 - Conference Track Proceedings*, International Conference on Learning Representations, ICLR.
 - [87] Horn, B. K. P., and Schunck, B. G., 1981, "Determining Optical Flow," *Artif. Intell.*, **17**(1–3), pp. 185–203.
 - [88] Østergaard Noe, K., De Senneville, B. D., Elstrøm, U. V., Tanderup, K., and Sørensen, T. S., 2008, "Acceleration and Validation of Optical Flow Based Deformable Registration for Image-Guided Radiotherapy," *Acta Oncol. (Madr.)*, **47**(7), pp. 1286–1293.
 - [89] Honegger, D., Meier, L., Tanskanen, P., and Pollefeys, M., 2013, "An Open

Source and Open Hardware Embedded Metric Optical Flow CMOS Camera for Indoor and Outdoor Applications,” *Proceedings - IEEE International Conference on Robotics and Automation*, pp. 1736–1741.

- [90] Chao, H., Gu, Y., and Napolitano, M., 2014, “A Survey of Optical Flow Techniques for Robotics Navigation Applications,” *J. Intell. Robot. Syst. Theory Appl.*, **73**(1–4), pp. 361–372.
- [91] Hartmann, C., Wang, J., Opristescu, D., and Volk, W., 2018, “Implementation and Evaluation of Optical Flow Methods for Two-Dimensional Deformation Measurement in Comparison to Digital Image Correlation,” *Opt. Lasers Eng.*, **107**, pp. 127–141.
- [92] Pock, T., Urschler, M., Zach, C., Beichel, R., and Bischof, H., 2007, “A Duality Based Algorithm for TV-L1-Optical-Flow Image Registration,” *Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, Springer Verlag, pp. 511–518.
- [93] Pérez, J. S., Meinhardt-Llopis, E., and Facciolo, G., 2013, “TV-L1 Optical Flow Estimation,” *Image Process. Line*, **2013**, pp. 137–150.
- [94] Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., and Bengio, Y., 2020, “Generative Adversarial Networks,” *Commun. ACM*, **63**(11), pp. 139–144.
- [95] Tschuchnig, M. E., Oostingh, G. J., and Gadermayr, M., 2020, “Generative Adversarial Networks in Digital Pathology: A Survey on Trends and Future Potential,” *Patterns*, **1**(6), p. 100089.
- [96] Isola, P., Zhu, J.-Y., Zhou, T., and Efros, A. A., 2016, “Image-to-Image Translation with Conditional Adversarial Networks.”
- [97] Kattenborn, T., Eichel, J., and Fassnacht, F. E., 2019, “Convolutional Neural Networks Enable Efficient, Accurate and Fine-Grained Segmentation of Plant Species and Communities from High-Resolution UAV Imagery,” *Sci. Rep.*, **9**(1), pp. 1–9.
- [98] Zhu, J. Y., Park, T., Isola, P., and Efros, A. A., 2017, “Unpaired Image-to-Image Translation Using Cycle-Consistent Adversarial Networks,” *Proceedings of the IEEE International Conference on Computer Vision*, Institute of Electrical and Electronics Engineers Inc., pp. 2242–2251.
- [99] Tschuchnig, M. E., Oostingh, G. J., and Gadermayr, M., 2020, “Erratum: Generative Adversarial Networks in Digital Pathology: A Survey on Trends and Future Potential (*Patterns* (2020) 1(6), (S2666389920301173), (10.1016/j.Patter.2020.100089)),” *Patterns*.
- [100] “Pix2Pix | TensorFlow Core” [Online]. Available: <https://www.tensorflow.org/tutorials/generative/pix2pix>. [Accessed: 24-May-2021].
- [101] “CycleGAN | TensorFlow Core” [Online]. Available: <https://www.tensorflow.org/tutorials/generative/cyclegan>. [Accessed: 24-

May-2021].

Appendix 1

Preprocessing Algorithms

Optical Flow Image Registration:

```
#!/usr/bin/env python
# coding: utf-8

# In[1]:

import cv2
import numpy as np
import matplotlib.pyplot as plt
import os
import glob
from skimage.transform import warp
one_seg_only=True
one_seg_rem=False
seg_img=False
from skimage import io,registration
img_list=[]
layers=['dapi','cd3']
#path1=os.path.join("D:\\","Users","Abu Bakr","Desktop","Raw images")
path1=os.path.join("D:\\","Raw Images")
if(os.path.exists(path1)==False):
    print("Path Not Found")
else:
    img_list=[x for x in sorted(glob.glob(os.path.join(path1,'he', '*.png'))
)]

path2=os.path.join("C:\\","Users","Abu Bakr","Desktop","tnbc_processed")
if(os.path.exists(path2)==False):
    os.makedirs(path2,exist_ok=False)

# In[2]:

from math import sqrt

COLORS = (
    (0,0,128),
    (25,25,112),
    (75,0,130),
    (0,0,139),
    (72,61,139),
    (153,50,204),
    (148,0,211),
    (138,43,226),
    (147,112,219)
```

```

)

def closest_color(centers):
    closest_color_ctrs=[]
    i=0
    for rgb in centers:
        r, g, b = rgb
        color_diffs_pctr = []
        for color in COLORS:
            cr, cg, cb = color
            color_diff = sqrt(abs(r - cr)**2 + abs(g - cg)**2 + abs(b - cb)*
*2)
            color_diffs_pctr.append((color_diff))
        closest_color_ctrs.append((min(color_diffs_pctr),i))
        i=i+1
    return min(closest_color_ctrs)[1]

# In[3]:

img_list

# In[4]:

img_store=0
for img_no in img_list:
    print(img_no)
    if(os.path.exists(os.path.join(path2,f"{img_store:03d}"))==False):
        os.makedirs(os.path.join(path2,f"{img_store:03d}"),exist_ok=False)
    img=cv2.imread(img_no)
    img=cv2.cvtColor(img,cv2.COLOR_BGR2RGB)
    pixel_values = img.reshape((-1, 3))
    pixel_values = np.float32(pixel_values)
    iterations=50
    epsilon_value=0.0001 #difference between clustered colours
    criteria = (cv2.TERM_CRITERIA_EPS + cv2.TERM_CRITERIA_MAX_ITER,iteration
s, epsilon_value)
    k = 5 #number of unique colors
    _, labels, (centers) = cv2.kmeans(pixel_values, k, None, criteria, 5, cv
2.KMEANS_RANDOM_CENTERS)
    # convert back to 8 bit values
    centers = np.uint8(centers)
    blue_idx=closest_color(centers)
    # flatten the labels array
    labels = labels.flatten()
    segmented_image = centers[labels.flatten()]
    segmented_image = segmented_image.reshape(img.shape)
    cv2.imwrite(os.path.join(path2,f"{img_store:03d}"),"segmented_image "+f"{
img_store:03d}"+".png"),segmented_image)

```

```

        temp = temp.reshape(img.shape)
        t2=cv2.cvtColor(temp,cv2.COLOR_RGB2BGR)
        # t2=cv2.medianBlur(t2,25)
        cv2.imwrite(os.path.join(path2,f"{img_store:03d}", "seg_one_rem "
+f"{i:03d}"+".png"),t2)
        img_store=img_store+1

# In[5]:

img_list_blue=[os.path.join(path2,f"{img_store:03d}", "blue_nuclei"+".png" )
for img_store in range(len(img_list))]

img_list_dapi=[x for x in sorted(glob.glob(os.path.join(path1,'dapi','*.png'
)) )]
img_list_cd3=[x for x in sorted(glob.glob(os.path.join(path1,'cd3','*.png')
))]
img_list_dapi=[x for x in sorted(glob.glob(os.path.join(path1,'dapi','tiles
','*.png')) )]
img_list_cd3=[x for x in sorted(glob.glob(os.path.join(path1,'cd3','tiles',
'*.png')) )]]
img_list_cd20=[x for x in sorted(glob.glob(os.path.join(path1,'cd20','*.png
')) )]

# In[6]:

img_store=0
for img1,img2 in zip(img_list_blue,img_list_dapi):
    path3=os.path.join(path2,f"{img_store:03d}",'align')
    if(os.path.exists(os.path.join(path3))==False):
        os.makedirs(os.path.join(path3),exist_ok=False)
    img1=cv2.imread(img1,cv2.IMREAD_GRAYSCALE)
    img2=cv2.imread(img2,cv2.IMREAD_GRAYSCALE)
    img1=cv2.resize(img1,(3328,3328))
    img2=cv2.resize(img2,(3328,3328))
    rh1,img1_1 = cv2.threshold(img1,0,255,cv2.THRESH_BINARY+cv2.THRESH_OTSU)
    rh2,img2_1 = cv2.threshold(img2,0,255,cv2.THRESH_BINARY+cv2.THRESH_OTSU)
    #img1_1=cv2.bitwise_not(img1_1)
    #img2_1=cv2.bitwise_not(img2_1)
    cv2.imwrite(os.path.join(path3,"img1 of"+f"{img_store:03d}"+".png"),img1
_1)
    cv2.imwrite(os.path.join(path3,"img2 of"+f"{img_store:03d}"+".png"),img2
_1)

    print("thresholding done for "+f"{img_store:03d}")
# --- Compute the optical flow
    #v, u = registration.optical_flow_tv11(img1_1, img2_1,attachment=5, tigh
tness=0.3, num_warp=5, num_iter=10, tol=0.0001)

```

```

v, u = registration.optical_flow_tv11(img1_1, img2_1)
print("alignment done for "+f"{img_store:03d}")
# --- Use the estimated optical flow for registration

nr, nc = img1.shape
row_coords, col_coords = np.meshgrid(np.arange(nr), np.arange(nc), indexing='ij')
image1_warp = warp(img2_1, np.array([row_coords + v, col_coords + u]), mode='nearest')
img1_rgb=cv2.cvtColor(img1_1,cv2.COLOR_GRAY2BGR)
image1_warp=image1_warp.astype(np.float32)
img2_rgb=cv2.cvtColor(image1_warp,cv2.COLOR_GRAY2BGR)
red = np.zeros((3328,3328,3),dtype=np.float32)
red[:]=(255,0,0)
blue = np.zeros((3328,3328,3),dtype=np.float32)
blue[:]=(0,0,255)
plt.imsave(os.path.join(path3,"overlap of"+f"{img_store:03d}"+".png"),cv2.bitwise_not(img1_rgb)*blue+
            cv2.bitwise_not(img2_rgb)*red)
plt.imsave(os.path.join(path3,"warp of"+f"{img_store:03d}"+".png"),img2_rgb,cmap='gray')
img_to_process=[]
img_to_process.append(img_list_dapi[img_store])
img_to_process.append(img_list_cd3[img_store])
#img_to_process.append(img_list_cd20[img_store])
processed_imgs=[]
for img2_main in img_to_process:
    img2_main=cv2.imread(img2_main)
    img2_main=cv2.resize(img2_main,(3328,3328))
    img2_r,img2_g,img2_b=cv2.split(img2_main)

    image1_r_warp = warp(img2_r, np.array([row_coords + v, col_coords + u]),
                        mode='nearest')
    plt.imsave(os.path.join(path3,"image_r of"+f"{img_store:03d}"+".png"),image1_r_warp,cmap='gray')
    image1_g_warp = warp(img2_g, np.array([row_coords + v, col_coords + u]),
                        mode='nearest')
    plt.imsave(os.path.join(path3,"image_g of"+f"{img_store:03d}"+".png"),image1_g_warp,cmap='gray')
    image1_b_warp = warp(img2_b, np.array([row_coords + v, col_coords + u]),
                        mode='nearest')
    plt.imsave(os.path.join(path3,"image_b of"+f"{img_store:03d}"+".png"),image1_b_warp,cmap='gray')
    img1=cv2.imread(os.path.join(path3,"image_r of"+f"{img_store:03d}"+".png"))
    img2=cv2.imread(os.path.join(path3,"image_g of"+f"{img_store:03d}"+".png"))

```

```

img3=cv2.imread(os.path.join(path3,"image_b of"+f"{img_store:03d}"+
.png"))
img1=cv2.cvtColor(img1,cv2.COLOR_BGR2GRAY)
img2=cv2.cvtColor(img2,cv2.COLOR_BGR2GRAY)
img3=cv2.cvtColor(img3,cv2.COLOR_BGR2GRAY)
np.multiply(img1, 1, out=img1, casting="unsafe")
np.multiply(img2, 1, out=img2, casting="unsafe")
np.multiply(img3, 1, out=img3, casting="unsafe")
after = cv2.merge([img1,img2,img3])
processed_imgs.append(after)
for img_present,toi in zip(processed_imgs,layers):
    cv2.imwrite(os.path.join(path3,toi+f"{img_store:03d}"+
),img_present)
    img_store=img_store+1

```

Overlay:

```

import cv2
import matplotlib.pyplot as plt
import numpy as np
from glob import glob

img=[x for x in sorted(glob("D:\\Preprocessed Imgs _8th Apr\\crc_raw\\he\\*.
.png"))]
#img=[x for x in sorted(glob("C:\\Users\\Abu Bakr\\Desktop\\Raw images\\he\\
*.png"))]
#img=[x for x in sorted(glob("D:\\crc\\he\\*.png"))]
j=0
for i in img:

    print(i)
    #background = cv2.imread("C:\\Users\\Abu Bakr\\Desktop\\Documents_folder
\\Raw Images\\"+str(i)+".png")
    background= cv2.imread(i)
    cv2.imwrite("C:\\Users\\Abu Bakr\\Desktop\\he_images\\"+f"{4+j:03d}"+".p
ng",background)
    overlay = cv2.imread("D:\\Preprocessed Imgs _8th Apr\\crc_processed\\"+
f"{j:03d}"+ "\\align\\cd3"+f"{j:03d}"+ ".png")
    #overlay=cv2.imread("C:\\Users\\Abu Bakr\\Desktop\\Documents_folder\\Pro
cessed Images\\"+str(j)+"\\align\\cd3"+str(j)+ ".png")
    background=cv2.resize(background,(3328,3328))
    overlay=cv2.cvtColor(overlay,cv2.COLOR_RGB2GRAY)
    zero=np.zeros_like(overlay)
    img2=cv2.merge([zero,overlay,zero])
    img2=cv2.GaussianBlur(img2,(255,255),0)
    #background=cv2.GaussianBlur(background,(25,25),0)
    added_image = cv2.addWeighted(background,0.3,img2,1,10)
    cv2.imwrite("C:\\Users\\Abu Bakr\\Desktop\\test3\\"+f"{4+j:03d}"+ ".png",
added_image)
    added_image = cv2.addWeighted(img2,6,background,0.3,0)
    #cv2.imwrite("C:\\Users\\Abu Bakr\\Desktop\\if\\tnbc\\test 2 "+str(i)+ ".
png",added_image)

```

```
cv2.imwrite("C:\\Users\\Abu Bakr\\Desktop\\test4\\"+f"{4+j:03d}"+".png",  
added_image)
```

```
j=j+1
```

Appendix 2

Deep Learning Algorithms

Pix2Pix GAN [100]:

Creation of Encoder Structure

```
def downsample(filters, size, apply_batchnorm=True):
    initializer = tf.random_normal_initializer(0., 0.02)

    result = tf.keras.Sequential()
    result.add(
        tf.keras.layers.Conv2D(filters, size, strides=2, padding='same',
                                kernel_initializer=initializer, use_bias=False)
    )

    if apply_batchnorm:
        result.add(tf.keras.layers.BatchNormalization())

    result.add(tf.keras.layers.LeakyReLU())

    return result

down_model = downsample(3, 4)
down_result = down_model(tf.expand_dims(inp, 0))
```

Creation of Decoder Structure

```
def upsample(filters, size, apply_dropout=False):
    initializer = tf.random_normal_initializer(0., 0.02)

    result = tf.keras.Sequential()
    result.add(
        tf.keras.layers.Conv2DTranspose(filters, size, strides=2,
                                         padding='same',
                                         kernel_initializer=initializer,
                                         use_bias=False))

    result.add(tf.keras.layers.BatchNormalization())

    if apply_dropout:
        result.add(tf.keras.layers.Dropout(0.5))

    result.add(tf.keras.layers.ReLU())

    return result

up_model = upsample(3, 4)
up_result = up_model(down_result)
```


Generator Structure

```
def Generator():
    size=4
    inputs = tf.keras.layers.Input(shape=[256,256,3])

    down_stack = [
        downsample(64, size, apply_batchnorm=False), # (bs, 128, 128, 64)
        downsample(128,size), # (bs, 64, 64, 128)
        downsample(256, size), # (bs, 32, 32, 256)
        downsample(512, size), # (bs, 16, 16, 512)
        downsample(512, size), # (bs, 8, 8, 512)
        downsample(512, size), # (bs, 4, 4, 512)
        downsample(512, size), # (bs, 2, 2, 512)
        downsample(512, size), # (bs, 1, 1, 512)
    ]

    up_stack = [
        upsample(512, size, apply_dropout=True), # (bs, 2, 2, 1024)
        upsample(512, size, apply_dropout=True), # (bs, 4, 4, 1024)
        upsample(512, size, apply_dropout=True), # (bs, 8, 8, 1024)
        upsample(512, size), # (bs, 16, 16, 1024)
        upsample(256, size), # (bs, 32, 32, 512)
        upsample(128, size), # (bs, 64, 64, 256)
        upsample(64, size), # (bs, 128, 128, 128)
    ]

    initializer = tf.random_normal_initializer(0., 0.02)
    last = tf.keras.layers.Conv2DTranspose(OUTPUT_CHANNELS, size,
                                             strides=2,
                                             padding='same',
                                             kernel_initializer=initializer,
                                             activation='tanh') # (bs, 256, 256,
3)

x = inputs

# Downsampling through the model
skips = []
for down in down_stack:
    x = down(x)
    skips.append(x)

skips = reversed(skips[:-1])

# Upsampling and establishing the skip connections
for up, skip in zip(up_stack, skips):
    x = up(x)
    x = tf.keras.layers.Concatenate()([x, skip])

x = last(x)
```

```
return tf.keras.Model(inputs=inputs, outputs=x)
```

Discriminator Structure

```
def Discriminator():
    size=4
    initializer = tf.random_normal_initializer(0., 0.02)

    inp = tf.keras.layers.Input(shape=[256, 256, 3], name='input_image')
    tar = tf.keras.layers.Input(shape=[256, 256, 3], name='target_image')

    x = tf.keras.layers.concatenate([inp, tar]) # (bs, 256, 256, channels*2)

    down1 = downsample(64, size, False)(x) # (bs, 128, 128, 64)
    down2 = downsample(128, size)(down1) # (bs, 64, 64, 128)
    down3 = downsample(256, size)(down2) # (bs, 32, 32, 256)

    zero_pad1 = tf.keras.layers.ZeroPadding2D()(down3) # (bs, 34, 34, 256)
    conv = tf.keras.layers.Conv2D(512, size, strides=1,
                                   kernel_initializer=initializer,
                                   use_bias=False)(zero_pad1) # (bs, 31, 31, 512)

    batchnorm1 = tf.keras.layers.BatchNormalization()(conv)

    leaky_relu = tf.keras.layers.LeakyReLU()(batchnorm1)

    zero_pad2 = tf.keras.layers.ZeroPadding2D()(leaky_relu) # (bs, 33, 33, 512)

    last = tf.keras.layers.Conv2D(1, size, strides=1,
                                   kernel_initializer=initializer)(zero_pad2) # (bs, 30, 30, 1)

    return tf.keras.Model(inputs=[inp, tar], outputs=last)
```

Train Step

```
@tf.function
def train_step(input_image, target, epoch):
    with tf.GradientTape() as gen_tape, tf.GradientTape() as disc_tape:
        gen_output = generator(input_image, training=True)

        ps=tf.reduce_mean(tf.image.psnr(gen_output,target, max_val=1.0))
        #tf.print(ps)
        #psnr.append(ps)
        ss=tf.reduce_mean(tf.image.ssim(gen_output,target, 1.0))
        #ssim.append(ss)
        #tf.print(ss)
        disc_real_output = discriminator([input_image, target], training=True)
```

```

        print()
        a,b,c,d,e,f=train_step(input_image, target, epoch)
        gen_total_loss_1.append(a)
        gen_gan_loss_1.append(b)
        gen_l1_loss_1.append(c)
        disc_loss_1.append(d)
        psnr.append(e)
        ssim.append(f)
    print()

    # saving (checkpoint) the model every 5 epochs
    if (epoch + 1) % 5 == 0:
        checkpoint.save(file_prefix = checkpoint_prefix)

    print ('Time taken for epoch {} is {} sec\n'.format(epoch + 1,
                                                         time.time()-start))
    checkpoint.save(file_prefix = checkpoint_prefix)

```

CycleGAN [101]:

Creation of Generators and Discriminators

```

OUTPUT_CHANNELS = 3

generator_g = pix2pix.unet_generator(OUTPUT_CHANNELS, norm_type='instancenorm')
generator_f = pix2pix.unet_generator(OUTPUT_CHANNELS, norm_type='instancenorm')

discriminator_x = pix2pix.discriminator(norm_type='instancenorm', target=False)
discriminator_y = pix2pix.discriminator(norm_type='instancenorm', target=False)

```

Training:

```

@tf.function
def train_step(real_x, real_y):
    # persistent is set to True because the tape is used more than
    # once to calculate the gradients.
    with tf.GradientTape(persistent=True) as tape:
        # Generator G translates X -> Y
        # Generator F translates Y -> X.

        fake_y = generator_g(real_x, training=True)
        cycled_x = generator_f(fake_y, training=True)

        fake_x = generator_f(real_y, training=True)
        cycled_y = generator_g(fake_x, training=True)

        # same_x and same_y are used for identity loss.

```

```

same_x = generator_f(real_x, training=True)
same_y = generator_g(real_y, training=True)

disc_real_x = discriminator_x(real_x, training=True)
disc_real_y = discriminator_y(real_y, training=True)

disc_fake_x = discriminator_x(fake_x, training=True)
disc_fake_y = discriminator_y(fake_y, training=True)

# calculate the loss
gen_g_loss = generator_loss(disc_fake_y)
gen_f_loss = generator_loss(disc_fake_x)

total_cycle_loss = calc_cycle_loss(real_x, cycled_x) + calc_cycle_loss(r
eal_y, cycled_y)

# Total generator loss = adversarial loss + cycle loss
total_gen_g_loss = gen_g_loss + total_cycle_loss + identity_loss(real_y,
same_y)
total_gen_f_loss = gen_f_loss + total_cycle_loss + identity_loss(real_x,
same_x)

disc_x_loss = discriminator_loss(disc_real_x, disc_fake_x)
disc_y_loss = discriminator_loss(disc_real_y, disc_fake_y)
ps=tf.reduce_mean(tf.image.psnr(fake_y,real_y, max_val=1.0))
#tf.print(ps)
#psnr.append(ps)
ss=tf.reduce_mean(tf.image.ssim(fake_y,real_y, 1.0))

# Calculate the gradients for generator and discriminator
generator_g_gradients = tape.gradient(total_gen_g_loss,
generator_g.trainable_variables)
generator_f_gradients = tape.gradient(total_gen_f_loss,
generator_f.trainable_variables)

discriminator_x_gradients = tape.gradient(disc_x_loss,
discriminator_x.trainable_variab
les)
discriminator_y_gradients = tape.gradient(disc_y_loss,
discriminator_y.trainable_variab
les)

# Apply the gradients to the optimizer
generator_g_optimizer.apply_gradients(zip(generator_g_gradients,
generator_g.trainable_variables)
)

generator_f_optimizer.apply_gradients(zip(generator_f_gradients,
generator_f.trainable_variables)
)

```

```

        discriminator_x_optimizer.apply_gradients(zip(discriminator_x_gradients,
                                                    discriminator_x.trainable_variables))

        discriminator_y_optimizer.apply_gradients(zip(discriminator_y_gradients,
                                                    discriminator_y.trainable_variables))
    return disc_x_loss, disc_y_loss, total_gen_f_loss, total_gen_g_loss, ps, ss

EPOCHS=10

for epoch in range(EPOCHS):
    start = time.time()
    l1=[]
    l2=[]
    l3=[]
    l4=[]
    l5=[]
    l6=[]
    n = 0
    for image_x, image_y in train_dataset:
        a,b,c,d,e,f=train_step(image_x, image_y)
        l1.append(a.numpy())
        l2.append(b.numpy())
        l3.append(c.numpy())
        l4.append(d.numpy())
        l5.append(e.numpy())
        l6.append(f.numpy())
        if n % 2 == 0:
            print('.', end='')
        n+=1

    clear_output(wait=True)
    # Using a consistent image (sample_horse) so that the progress of the mode
1    # is clearly visible.
    a1=(sum(l1)/len(l1))
    a2=(sum(l2)/len(l2))
    a3=(sum(l3)/len(l3))
    a4=(sum(l4)/len(l4))
    a5=(sum(l5)/len(l5))
    a6=(sum(l6)/len(l6))
    print("\nDisc_x_loss "+str(a1))
    print("\nDisc_y_loss "+str(a2))
    print("\nTotal_gen_f_loss "+str(a3))
    print("\nTotal_gen_g_loss "+str(a4))
    disc_x_loss.append(a1)
    disc_y_loss.append(a2)
    total_gen_f_loss.append(a3)

```

```

total_gen_g_loss.append(a4)
psnr.append(a5)
ssim.append(a6)
generate_images(generator_g, sample_he, sample_if)

if (epoch + 1) % 5 == 0:
    ckpt_save_path = ckpt_manager.save()
    print ('Saving checkpoint for epoch {} at {}'.format(epoch+1,
                                                         ckpt_save_path))

print ('Time taken for epoch {} is {} sec\n'.format(epoch + 1,
                                                    time.time()-start))

```