Sketch Recognition based on Deep Learning

for Children's Psychological Development Monitoring

by

Fatin Nur Syafiqah binti Shamrulismawi

17004154

Dissertation submitted in partial fulfilment of

the requirements for the

BACHELOR OF INFORMATION SYSTEM

SEPTEMBER 2021

Universiti Teknologi PETRONAS

Seri Iskandar

32610 Tronoh

Perak Darul Ridzuan

CERTIFICATION OF APPROVAL

Sketch Recognition based on Deep Learning

for Children's Psychological Development Monitoring

by

Fatin Nur Syafiqah binti

Shamrulismawi

17004154

A project dissertation submitted to the

Information System Programme

Universiti Teknologi PETRONAS

in partial fulfilment of the requirement for the

BACHELOR OF INFORMATION SYSTEM

(Dr Mohd Nordin B Zakaria)

UNIVERSITI TEKNOLOGI PETRONAS

SERI ISKANDAR, PERAK

September 2021

CERTIFICATION OF ORIGINALITY

This is to certify that I am responsible for the work submitted in this project, that the original work is my own except as specified in the references and acknowledgements, and that the original work contained herein have not been undertaken or done by unspecified sources or persons.

fatin

____

FATIN NUR SYAFIQAH BINTI SHAMRULISMAWI

ABSTRACT

Beyond aesthetic and educational purposes, children's drawings have been recognized throughout the past century as markers of cognitive development as well as personality, emotional and social functioning in adolescents. With the continuous development of globalization along with civilization, mental health concerns are receiving greater attention as a result. Draw-A-Person (DAP) and Draw-a-Family-Picture-Test (DAFPT) are one of the conventional psychological assessment methods based around children’s drawings, where the assessments are meant to assist experts in determining a child's cognitive development level with little to no influence from other outlier variables, such as language difficulties and special needs. However, to identify children's development progress manually and solely by specialists is an obsolete approach, especially in light of the technological developments that have occurred in recent years. The lack of psychological identification tool on children's emotional and mental growth, and limited studies that utilized sketch recognition for children's drawings have suggested a solution in employing Artificial Intelligence (AI) in automating such procedures. In view of its vast psychological use, this study presents a sketch recognizer engine that is geared towards children's drawings, where it will be developed using the Rapid Application Development (RAD) methodology. Age-specific identification will be built on existing Deep Learning technologies, especially Convolutional Neural Network (CNN). The resultant tool is intended to open the way for a huge amount of study in children's psychological development in the future, such as virtually detecting emotions through their drawings.

ACKNOWLEDGEMENT

My appreciation and gratitude are extended towards Universiti Teknologi PETRONAS for the given opportunity to develop and complete this sketch recognition engine as part of my Final Year Project (FYP).

All praises to Allah SWT for His guidance, which has aided me through the challenging times by providing me with the strength and motivation to complete this project and dissertation. As a result, I'd like to express my sincere gratitude to my project supervisor, Dr. Mohd Nordin B Zakaria, for his guidance and unwavering support throughout this journey to project completion. I am grateful for every piece of advice and assistance given, especially on how to use the coding optimally and support whenever I encounter problems. Aside from that, I'd like to thank Dr. Siti Rohkmah Mohd Shukri of Monash University Malaysia for facilitating this project collaboration and assisting throughout the project's planning and requirement phases.

Other than that, I would like to express my appreciation towards my parents and family members for the overwhelming moral support throughout the entire project. Without them, I would not be where I am today, as an individual and a graduating final year student. All in all, thank you to everyone who played a role throughout my Final Year Project experience. I am eternally grateful for all the support and guidance I have received in successfully completing this Sketch Recognition based on Deep Learning for Children’s Psychological Development Monitoring project.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ABBREVIATIONS AND NOMENCLATURES

DAP = Draw-A-Person

DAFPT = Draw-a-Family-Picture-Test

AI = Artificial Intelligence

RAD = Rapid Application Development

CNN = Convolutional Neural Network

RNNs = Recurrent Neural Networks

HTP = House-Tree-Person

KFD = Kinetic Family Drawings

CDT = Clock-Drawing Tests

RAD = Rapid Application Development

CHAPTER 1

INTRODUCTION

1. INTRODUCTION

1.1 Background of Study

In childhood, developmental delays are prevalent, affecting 10% to 15% of pre-schoolers (Choo, Agarwal, How & Yeleswarapu, 2019). It is a condition whereby children fail to reach developmental milestones in contrast to peers of the same age range. The degree of developmental delay can be further categorized as mild (functional age with 33 % below chronological age), moderate (functional age with 34% to 66% of chronological age), and severe (functional age with 66% below chronological age). Developmental delays are commonly identified through manual monitoring by parents or by experts. Nevertheless, children's drawings have been extensively acknowledged throughout the past century as a useful tool for assessing cognitive development levels. These results illustrate how drawing skills are considered to be an indication of children's cognitive, emotional, and social functioning, where they are the windows to children's mind. Therefore, while the manual monitoring and identification processes by experts is regarded to be an obsolete approach in this fast-paced technological world, children's drawings, on the other hand, can be used to facilitate the process of identifying their psychological development growth as well as eliminate developmental delays in children through early childhood detection.

Children's drawings, indicators of cognitive development as well as personality, emotional and social functioning offers such significance in enhancing the identification process of children's psychological growth. Accordingly, the combination and iteration process of adapting Artificial Intelligence (AI) into a real world problem along with a well-known psychological concept on children's drawings will not only reduce human errors in identification processes, but also enhance the pace of detection as it can be administered by non-professionals, such as parents or teachers. AI is a simulation of human intelligence which allows machines to rationalize and select the best actions to take in order to achieve a given goal, much as humans do in real life. As a result, diagnostic choices and treatment results can be improved. The use of AI in medical imaging and diagnostics has grown considerably. Diagnostic mistakes can be prevented, and test results can be improved with the use of deep learning algorithms. The study of machine learning and neural networks are amongst one of the important disciplines of AI. One of the most prevalent AI-embedded technology is image recognition with its applications ranging from personal photo management to visual search engines, which suggests that the contemporary global community is familiar with this technology and is accustomed to its use.

Deep Learning is a further subset of Machine Learning and AI, where it allows computers to make human-like judgments using their own neural networks. It employs many layers to extract higher-level characteristics from raw input, which is based on biological nodes in the human body, allowing computers to identify and interpret images timely and efficiently. Machine Learning and Deep Learning are phrases that are occasionally used interchangeably. It is important to note, however, that both models have quite distinct competences. When it comes to determining an accurate prediction of image recognition, machine learning would occasionally need human guidance as the trained data might be inadequate, whereas for deep learning, its algorithm is able to determine on its own if a prediction is accurate or not due to its own neural network. As a result of this neural network, deep learning is recognized as the most human-like technology since it can learn on its own by developing its own computing technique. Subsequently, deep learning will allow the process of detecting age-specific identification of children's psychological growth to be easier and faster. In this paper, Convolutional Neural Network (CNN), a class of deep learning methods, will be used to evaluate graph structure data of children's drawings as it offers a better way of dealing with abstract concepts and simplifying the problems into simpler representations from different perspectives. Hence, this framework will allow better training of the dataset and prediction on whether the age of a child's drawing is equivalent to the real age, as well as to predict potential developmental delays on children by image recognition.

1.2 Problem Statement

Current child-centred psychological researchers have led to the development of a variety of novel ways for gaining access to children's viewpoints, which paves way to detecting their psychological growth. In the history of developmental psychology, drawings were used a lot, although they were more commonly employed as projective assessments than as a tool to access the perspectives of the children. Therefore, this section outlines the inefficiency of manual identification by experts, which is supported by the lack of psychological identification tool on children's emotional and mental growth, limited studies that utilized sketch recognition for children's drawings and the existing sketch recognition techniques are considerably static.

i. **There are limited studies that utilized sketch recognition for children's drawings.**

   The prevalence of touchscreen devices has made drawing data collection easier than ever before, due to their wide availability. Computers are now able to detect human drawings using sketch recognition, rapidly with the advent of deep learning algorithms. Thus, drawings are now widely utilized in domains such as human-computer interface and computer vision recognition because of its versatility and flexibility. However, there are still limited studies that used sketch recognition specifically for children's drawings (Kim, Taele, Seo, Liew, & Hammond, 2016). This is due to the fact that most research were done exclusively on drawings from older subjects, with no technological methods or algorithms used in the process of analysis. Therefore, this paper aims to elucidate on a development of a sketch recognition tool focusing on children's drawing, which would cover extensive research on the needs towards its development.

ii. **Lack of psychological identification tool on children's emotional and mental growth.**

   There is a significant prevalence of mental health problems among children rising recently. In absence of well-validated psychological identification tool that can be used across contexts, there is a definite scarcity of resources that can be used (Nezafat, Chandna, & Gladstone, 2019). Subsequently, in this modern society, the implementation of system intelligence and computer vision through deep learning will help to prevent such mental issues developed in children, as it can be equipped and utilized easily. Though children’s drawings have been acknowledged as psychological indicators over the past century, there happens to be a limitation on gaining easy access to detecting negative or positive psychological development. Existing methods include consulting with specialists, which may not be feasible nor affordable to most families. There are also the existing risks of human error in diagnostics. Therefore, the aim to have an image recognition as a system which automates the process of age-specific psychological growth detection that is highly accessible and accurate, will address the issue of lack of psychological identification tool on children's emotional and mental growth. This is to prove how the modern technology can increase the efficacy in detecting cognitive and mental developments.

iii. The existing sketch recognition techniques are considerably static.

In virtue of the sparsity of signals and the high level of abstraction of drawings, learning meaningful representations of freehand sketches remains a challenge. Therefore, a novel way to describe drawings is proposed in this paper by the usage of multiple sparsely linked graphs, Convolutional Neural Network (CNN). This is due to the fact that CNN is widely used in pattern- and image-recognition problems because it has a number of advantages over other techniques such as Recurrent Neural Networks (RNNs) (Hijazi, Kumar, & Rowen, 2015). CNN not only can leverage the static and temporal sequence of sketches, but also be able to distinguish different graph structures by mapping them into different representations. Accordingly, CNNs are deep learning algorithms which extract features from dataset images by combining them with filters or kernels. In other words, it offers higher flexibility and accuracy in image recognition and feature extraction as opposed to existing sketch recognition techniques.

1.3 Objectives

The goal of this research project is to develop a tool which automates the process of identifying children's psychological development growth through image recognition technology and deep learning method. This tool will be able to aid the users, which are parents or teachers in detecting children’s psychological growth, whether the cognitive age is equivalent to their respective actual age and take measurable actions in preventing any psychological dysfunctions and developmental delays through early childhood detection. Following are the objectives of this research which addresses the stated problem statements, in order to achieve the mentioned goals:

- To conduct an extensive research on sketch recognition tuned towards children’s drawings.
- To implement a deep learning training model trained on a dataset of children's drawings.
- To develop a sketch recognition tool engine which automates the process of identifying children's psychological development growth with CNN.

1.4 Scope of Study

The scope of study is emphasized on elucidating on the areas of the research that will be investigated, which are the below outlined criterias this project will operate on. Due to the time constraint to develop a deep learning-based solution on a real-world problem, the scope of the study is done based on the feasibility of the project development and developer’s capabilities.

Targeted users

The targeted users of this sketch recognition tool which can be administered by non-professionals in replacements of psychological experts, are parents and teachers.

Methods and Tools

This project presents a sketch recognizer that is geared towards children's drawings through the Rapid Application Development (RAD) methodology that will monitor these operations phase by phase. Age-specific identification will be built on existing Deep Learning technologies, especially Convolutional Neural Network (CNN).

CHAPTER 2

LITERATURE REVIEW

2. LITERATURE REVIEW

In recent years, deep learning gives a rise to new wave in artificial neural network research, where it is commonly witnessed through various day-to-day business practices in utilizing image and facial recognitions. The nature of deep learning is a self-learning algorithm that builds a multilayer model and trains it with vast amounts of data (Meiyin Wu & Li Chen, 2015). Figure 2.1 shows the subsets of Artificial Intelligence, which contains Machine Learning, Deep Learning and lastly, Neural Networks. Artificial neural networks in this context, are interconnected groups of nodes inspired by the simplicity of neurons in the human brain. With the composition of such capabilities, very complex functions can be applied and improve the accuracy rate of the classification or prediction. Accordingly, human visual performance much outperforms computer capabilities, possibly due to greater high-level picture interpretation, contextual information, and massively parallel processing. Therefore, an AI-based solution incorporating both, deep learning, and neural networks for image recognition will be highly beneficial for its users. Relevant research studies on the matter have suggested that drawings are a depiction of intellect, where psychological growth may be detected at early age (Malchiodi, 1998). Accordingly, in evaluating the accuracy of the psychological drawing detection, psychological drawing tests are used as the existing conventional approach by specialists. Implementing an old research discovery of children drawings as indicators of their psychological growth with newly discovered deep learning neural network techniques would result into an image recognition tool with profound significance.

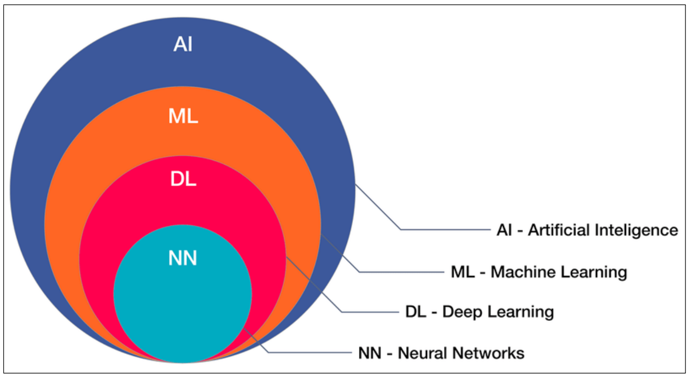

FIGURE 2.1: Subsets of Artificial Intelligence

2.1. Depiction of Intellect.

Children of chronological age have certain limited motor and cognitive skills that affect the aesthetics outcome of their drawings (Strommen, 1988). The depictions from their drawings can show various level of abstraction and distortion compared to drawings from older subjects, thus making the modelling of sketches unique and far more challenging. Children's drawings reflect their inner worlds, depicting various feelings, and relating information concerning psychological status and interpersonal style (Malchiodi, 1998). Although children may use drawings to explore or give visual form to ideas and observations, the overall consensus is that art expressions are uniquely distinct personal statements with components of conscious as well as unconscious significance, they can be indicative of many different qualities of the children who produced them. This is why children's drawings are a depiction of intellect, not only that, but the literature also supports the project feasibility in detecting their psychological growth through drawings. Accordingly, most therapists who work with children highly recognize that drawing is an effective therapeutic modality because it may help children express themselves in ways language cannot (Malchiodi, 1998). In depicting intelligence, one cannot only from a single perspective, whether it be on how fast they could speak or walk, however, children's drawings are proven to be one of the most concrete concepts in gaining access to a child's mind. Malchiodi (1998) also states that the study of children's drawings actually has quite a long tradition in the fields of psychiatry, psychology, art therapy and education. This long-standing fascination with children's art has generated a great deal of psychological drawings tests that have been generated and developed in evaluating a child's psychological development growth.

2.2. Psychological Drawing Tests.

Children's drawing has already been recognized over the past century. Apart from aesthetic and education reason, children's drawing has continuously been used among psychologists, psychotherapists, and social workers. For example, Draw-a-Person (DAP), House-Tree-Person (HTP) and Kinetic Family Drawings (KFD) Tests are used to evaluate traits such as cognitive, personality, emotional and social functioning of children (Leibowitz, 1999). Others such as Clock-Drawing Tests (CDT) can be used to screen cognitive impairment or spatial dysfunction and neglect. DAP is one of the most popular psychological drawing tests that is used to refined assessment of psychological maturity, where researchers found that evaluating the level of development of human figure drawings and measuring the number of features depicted provide successful indexes of children's levels of cognitive development and psychometric intelligence (Laak, Goede, Aleva & Rijswijk, 2005)

As a result, children's drawings serve as an indicator of their personal cognitive capacity, reflecting their intellectual abilities and progress. Florence Goodenough scientifically demonstrated this hypothesis, believing that the more cognitively developed a child is, the more realistic features he or she would add on a human figure drawing (Abell, Von Briesen & Watz, 1996). Afterwards, he then developed the DAP Test, which is currently one of the most popular psychological drawing tests for children that is used by many psychologists. According to the findings, human figure drawings are strongly and positively related to an individual's IQ. Subsequently, this shows that the necessity of interpreting children's drawing using sketch recognition can be tremendously beneficial with proven theories. This has motivated the development of the recognition tool as there has not been any main work that focuses on children's drawings data so far.

2.3. Neural Network Technique.

The prevalence of touchscreen devices has made acquiring sketch data much easier than ever. With the rapid development of deep learning techniques, computer these days are able to recognize human drawings using sketch recognition resulting in significant performance improvements. Existing techniques of sketch recognition are built upon Recurrent Neural Network (RNN) that focuses on the static nature of sketches and the temporal sequential property of sketches. A novel technique, Convolutional Neural Network (CNN) was introduced that uses multiple sparse graphs which simultaneously capture the stroke geometry, as well as temporal information of the sketches (Xu, Joshi & Bresson, 2019). CNN is a type of Artificial Neural Network (ANN), where it comprises and relates inputs to outputs with hidden layers of nodes. Figure 2.2 depicts how deep learning employs hidden layers as nodes, sometimes known as neurons since they mimic the learning process that happens in the brain. Each of these neurons receives a vast number of inputs, computes a weighted total, then passes them through an activation function and responds with an output, as shown in Figure 2.2. Without these neurons or hidden layers, as illustrated in Figure 2.3, the neural network is unsuitable for deep learning and has a poor rate of prediction accuracy. Instead of just one layer, CNN methods generally include image classification, grouping based on similarities discovered in image inputs, and performing image recognition, in this case, sketch recognition.

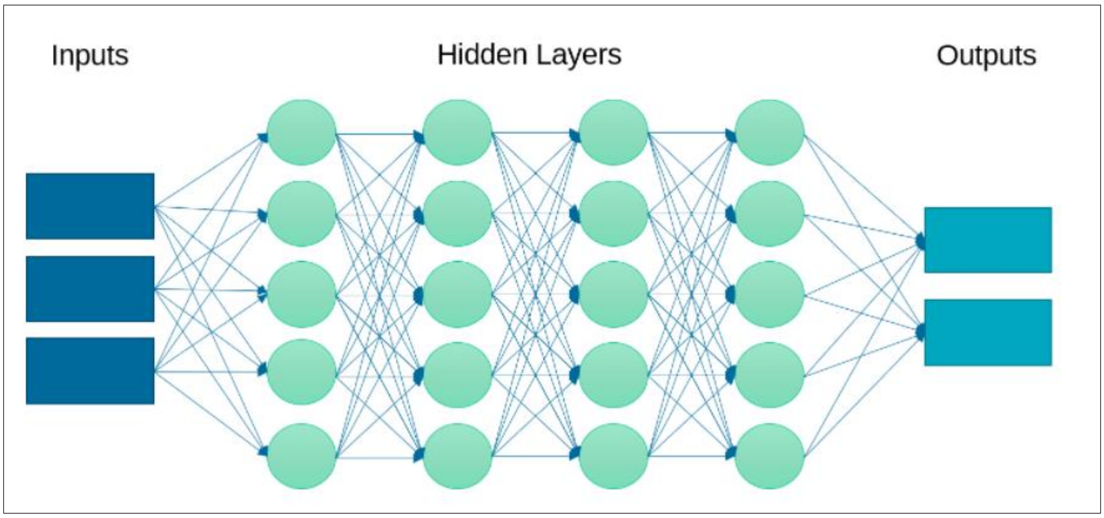

FIGURE 2.2: Multi-layer Neural Network

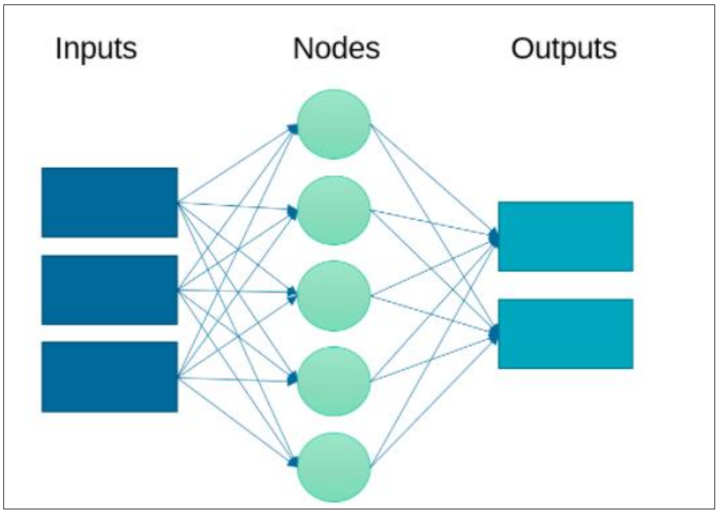

FIGURE 2.3: Single-layer Neural Network

For example, Li, Cai, Wang, Zhou, Feng, and Chen (2014) demonstrate how neural networks, especially CNN, are used in the medical business for lung image patch classification. Due to the scarcity of articles pertaining to CNN and children's drawings, this journal is used as an example of how a completely autonomous neural-based machine learning framework is built to extract discriminative features from training samples while also performing classification. CNN is a supervised learning technique, therefore higher classification results are to be expected because neural networks rely on training data to learn and increase their accuracy over time. Despite the promising approach, the study in does not directly address children's artwork. As a result, we want to advance the usage of the CNN approach in our sketch identification tool. The CNN that will be created will be able to recognise people based on their age. As a result, the CNN model will be able to operate with input classes and elements that are not often handled.

CHAPTER 3

METHODOLOGY

3. RESEARCH METHODOLOGY

For this research, the Rapid Application Development (RAD) is chosen to achieve the proposed goals for developing Sketch Recognition Tool on Children's Psychological Development Growth through Deep Learning. RAD is a common agile project management method in development processes. The main advantage of a RAD method is that this is a quick project under 1 year, where it is made feasible by eliminating the planning stage and emphasising prototype development. This leads to increased efficiency, quicker development, and more effective communication. Other benefits of this methodology are as below:

- Greater flexibility and agility, allowing them to make fast changes during the development process.
- The iterative technique shortens development time and speeds up project execution.

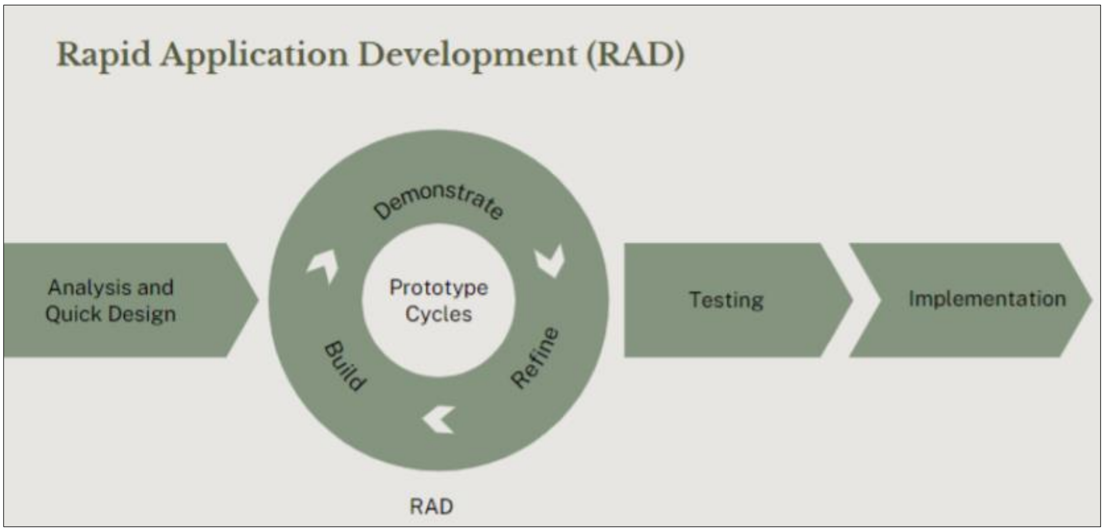

FIGURE 3.1: Rapid Application Development (RAD)

3.1 Project Activities

3.1.1 Analysis and Quick Design

In this phase, the project requirements and goals are analysed and designed. This step is essential in creating the flow for future phases as it dictates what is required to be developed. Clear and specific goals enable the project to be developed in a way that is consistent. The designs in this phase are not too complex as RAD emphasizes more on creating and iterating rather than planning. Once the goals are clear and the deadlines and milestones are set, the next phase can begin.

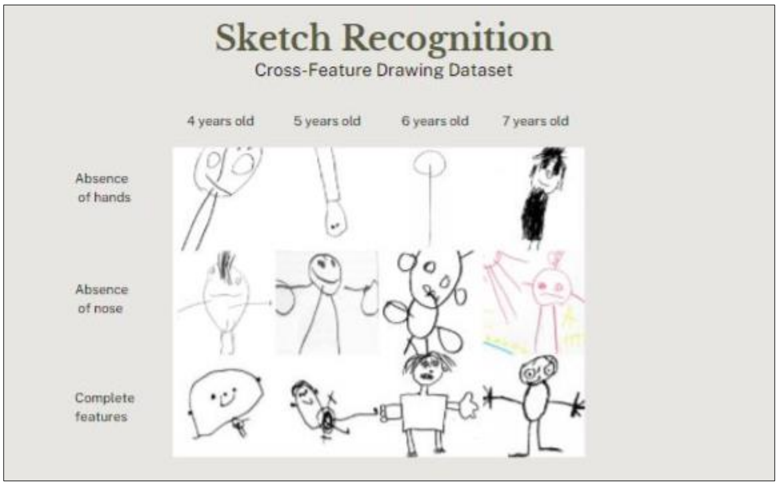

FIGURE 3.2: Cross-Feature Drawing Dataset

The cross-feature drawing dataset above shows the analysis of each children drawings obtained by cross-referencing the facial feature to the age. The analysis process is done prior to the training and testing of the dataset as it offers significance in viewing the appropriate data that is used. The sketch recognition will detect children's psychological developmental growth by extracting visual processing information from the temporal and order of their sketches. The children's drawings have been collected into a dataset of 1051 images and has been completed with image notations, as shown in FIGURE 3.2 above.

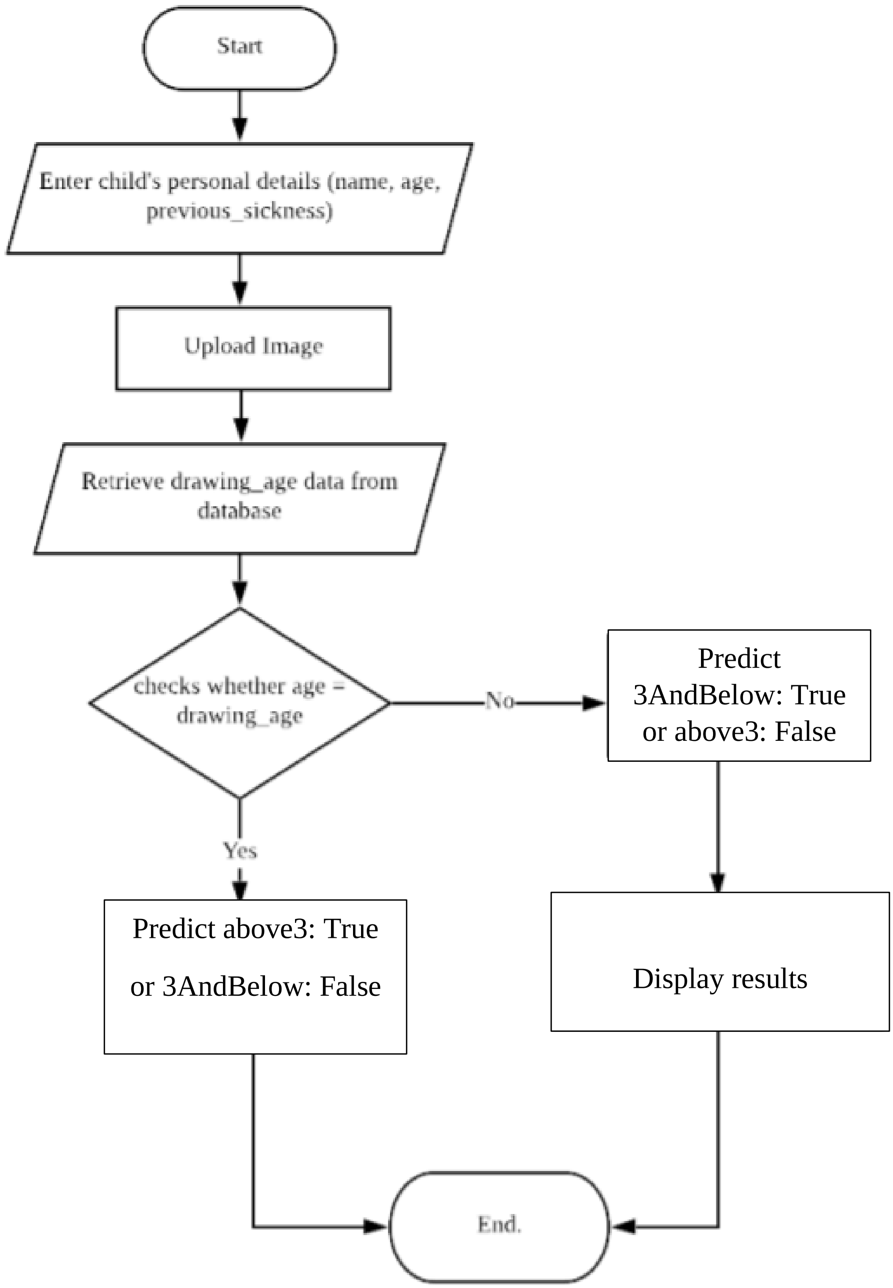

FIGURE 3.3: Project Flowchart

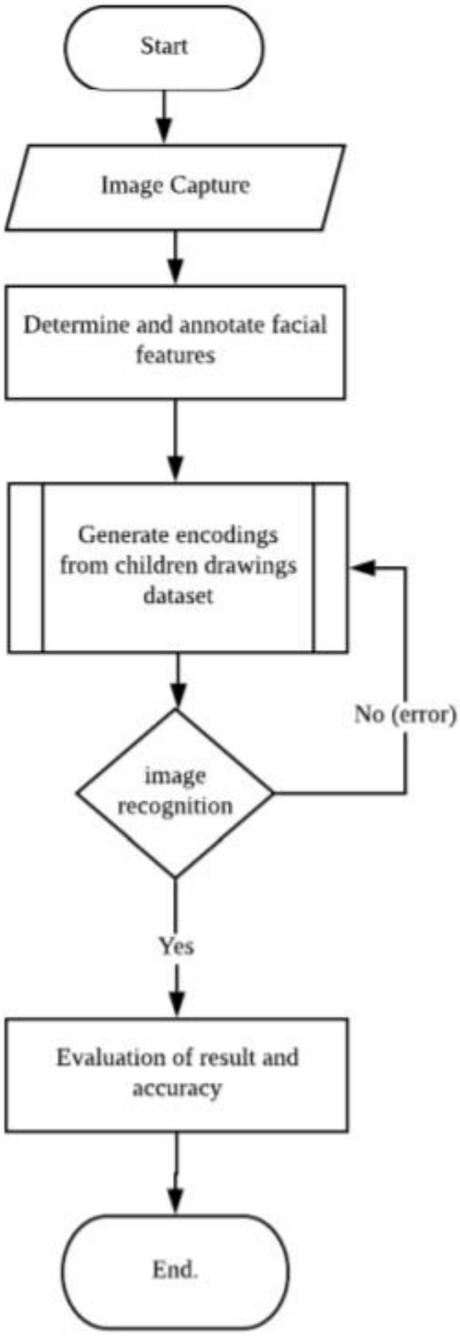

FIGURE 3.4: Image Recognition Flowchart

3.1.2 Prototype Cycles

Development cycles begin once goals and requirements have been defined. RAD prototyping cycles through three phases:

1. Build: The initial prototype is built in line with the requirements. Requirements were documented in an informal way of taking notes and minutes of meeting. A good communicator during the requirements exchange process is the priority because the requirement for deep learning needs to be as specific as possible.
2. Demonstrate: The prototype is demonstrated to the project lead or client and feedback on improvements in received. Project feedback was included in this phase for the preparation of the next cycle.
3. Refine: The feedback is received, and the prototype is further refined to suit prototype requirements.

This cycle continues until a satisfactory product is achieved. Using this method, create a minimum feature at first and then iterate and improve on them as development goes. This method allows developers to focus and tune their features more efficiently rather than taking a long time to plan and replan as development are being conducted continuously. Moreover, Requirements needs to be as specific as possible so that the prototype cycles continue to run smoothly as usual.

In this project, requirements were gathered through the collaboration with Dr. Siti Rohkmah Mohd Shukri of Monash University Malaysia. Responds and preparing for the risks are a continuous process in this phase which includes demonstrating, refining, and building the sketch recognition’s core engine. Communication is a crucial aspect in this cycle because improving the prototype is the priority with respect to the aligned requirements. The prototype cycles were conducted in fast pace because this project has the time planned for about 6 months.

3.1.3 Testing

Once a minimum viable product (MVP) is achieved, the product is then demonstrated to the client. Here the client inspects the solution and ensure all requirements and planned functionalities have been met. In this phase, third-party solutions are also integrated to ensure that the created product can function as intended with other application. Once the project has been thoroughly inspected and is deemed satisfactory, then implementation can be conducted.

Testing in this project involves pre-train tests using CUDA uses the same function as CPU tensors. However, it utilizes GPU for any computation. This pre-train process is to initialize the weights randomly in the neural network. When the pre-train starts, the weights are changed so that it can perform deep learning task with less mistakes.

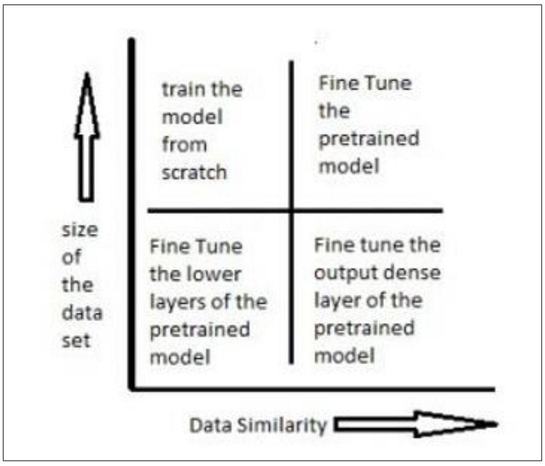

FIGURE 3.5: Pretrained Data Model

Data similarity in the prototype model increases every process of training a dataset towards the model. The process of identifying the correct weights for the neural network model with multiple backward and forward iteration is more accurate in which it improves the results of the algorithm. The pre-trained data networks have a strong ability to affect the whole prototype and algorithm due to its dependency to other functions in the algorithm.

3.1.4 Implementation

Implementation phase is the final phase of the RAD development. This phase occurs when all developments have been completed and the product is ready to be shipped. This phase includes the conversion of test data into real data, implementing the model in new systems, and training users on how to integrate the system in their work. No major changes are made in this phase as it should already be finalized. Any updates during this phase are catered towards fixes on any errors or bugs encountered during the product's use.

Optimization of the algorithm in multiple aspects can be done to improve stability, maintainability, and readability of the code. In this phase, this project seen the results of the algorithm, and the conclusion made for the specified dataset at the output of the algorithm. Connecting the back end, which is the algorithm, to real production data, developing a documentation, and maintenance tasks before finalizing the product can be done in this phase. The figure below shows the implementation of the algorithm towards real data where it can determine which classes the data associated to.

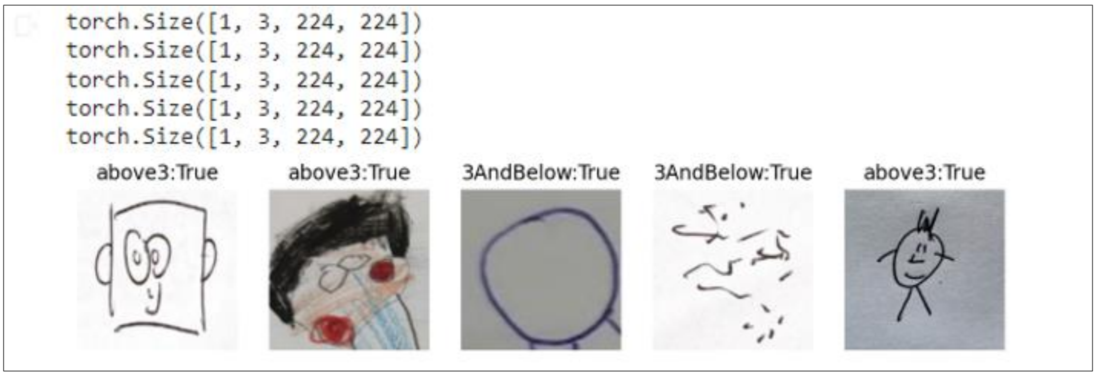

FIGURE 3.6: Results of algorithm

3.2 Project Milestone

The overall project milestones are executed and completed within the FYP 2 time frame, which can be seen as below:

TABLE 3.1: Project Milestone

Task Name	Start	End	Duration (days)
Project Topic Selection & Research	3/5/2021	16/5/2021	5
Requirement Gathering	17/5/2021	25/5/2021	8
Literature Review Research	26/5/2021	20/6/2021	26
Writing Algorithm Specification	21/6/2021	1/7/2021	10
Creating Prototype Designs	1/7/2021	10/7/2021	10
Find and Create Suitable Dataset	11/7/2021	1/9/2021	51
Developing Backend	2/9/2021	12/10/2021	40
Results Review	27/10/2021	1/11/2021	5
Dissertation	2/11/2021	29/11/2021	27

3.3 Programming Language

In this project, Python is the programming language, while utilizing **Google Colab** as the platform to code. The justification to choose Python as the programming language is as follows:

a) Extensive sets of library and framework selections

- Python is well-known to be the best programming language for machine learning (ML), artificial intelligence (AI), neural network (NN), and deep learning. Neural network learns to make predictions using a step that uses AI and ML which requires various types of libraries. Therefore, developers can develop a product faster without building an algorithm from scratch. In this project, various libraries are used to perform deep learning.

b) Platform independence

- In this project, the Python code can be a standalone algorithm that can be ran any platforms or operating system. Since it is independent, the algorithm can be easily distributed and executed which is suitable for this project because this project implementing Rapid Application Development (RAD). Data training in Python is much cheaper and faster compared to other programming languages. In this project, there was constant communication activity with the client where they were required to also run the algorithm by themselves, therefore the traits of independence in Python are suitable for the process.

c) Easy to read

- Python code can be read easily because it does not contain most of difficult characters or syntax in developing any functions. In this project, there are a lot of functions that needs good performance from other functions to run the algorithm smoothly. The easier the code to be understood, the faster the development. Therefore, the readability of the code determines the pace of the algorithm development.

CHAPTER 4

RESULTS AND DISCUSSION

4. RESULTS AND DISCUSSION

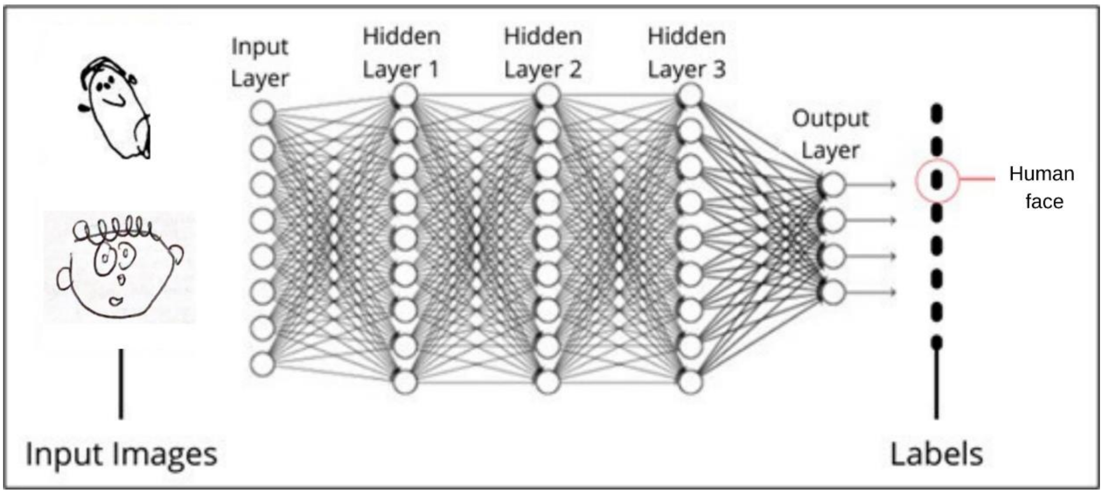

FIGURE 4.1: Deep Learning Layers of Growth Atlas

The results of this discussion are geared towards the project completion, which is the core engine for the sketch recognition for children's psychological development monitoring and are utilised to solve the issues specified in the problem statements and meet the previously stated objectives. The inefficiency of manual identification by experts are supported by facts that there are limited studies that utilized sketch recognition specifically for children's drawings, lack of psychological identification tool on children's emotional and mental growth and that the existing sketch recognition techniques are considerably static. In clarification, this project is more focused towards building and developing the core engine of the sketch recognizer by utilizing the CNN layers as shown in Figure 4.1 in predicting the outcomes, whether the age of the child is under or above 3 years old. 3 years old is used as an indicator for the project as it is part of a child's critical periods, where the number of connections between brain cells doubles than an adult (Sriram, 2020). This implies that the connections between a child's brain cells allow for quicker learning at this age than at any other period in life. As a result, early diagnosis of developmental delays during these critical periods would be tremendously useful.

4.1 There are limited studies that utilized sketch recognition for children's drawings.

The objective to be achieved for this problem statement is to conduct an extensive research on sketch recognition tuned towards children's drawings. After carefully researching and analysing previous project or research on the matter of monitoring psychological development growth by utilizing children's drawings with deep learning and neural networks, it has come to the attention that no such project with this innovation has existed. However, literatures have supported the theory separately, where children's drawings are good markers of cognitive development as well as personality, emotional and social functioning in children. Accordingly, previous deep learning research and projects has provided similarity in inspiring the development of this sketch recognizer, in terms of image classification and image recognition. By utilizing these two concepts, this project is built and completed successfully with 89% of accuracy. Though the reliability of judgments of children's social and emotional development and personality based on their drawings would be insufficient, this recognizer would act as the initial step in monitoring such delays, which would pave more ways for huge amount of study in children's psychological development in the future, such as virtually detecting emotions through their drawings and indicating the possible psychological dysfunctions of the child.

4.2 Lack of psychological identification tool on children's emotional and mental growth.

Accordingly, as there is significant prevalence of mental health problems among children rising recently, an identification tool that would aid in the process of early childhood developmental growth monitoring is crucial. The objective achieved to address this problem statement is to implement a deep learning training model trained on a dataset of children's drawings.

In building the deep learning training model, firstly is to organize the training dataset accordingly, which is done in Google Colab by utilizing python as the programming language and Pytorch as the machine learning library. After importing the modules as shown in the Figure 4.2 below, the train dataset loader is defined by utilizing the SubsetRandomSampler for the dataset split, as shown in Figure 4.3.

```
[ ]  %matplotlib inline
     %config InlineBackend.figure_format = 'retina'
     import matplotlib.pyplot as plt
     import numpy as np
     import torch
     from torch import nn
     from torch import optim
     import torch.nn.functional as F
     from torch.autograd import Variable
     from torchvision import datasets, transforms, models
```

FIGURE 4.2: Importing modules

```
    train_sampler = SubsetRandomSampler(train_idx)
    test_sampler = SubsetRandomSampler(test_idx)
    trainloader = torch.utils.data.DataLoader(train_data,
                   sampler=train_sampler, batch_size=64)
    testloader = torch.utils.data.DataLoader(test_data,
                   sampler=test_sampler, batch_size=64)
    return trainloader, testloader

trainloader, testloader = load_split_train_test(data_dir, .2)
print(trainloader.dataset.classes)
```

```
.  ..  .config  data  sample_data
loading data at  /content/data/train
train_data:  Dataset ImageFolder
    Number of datapoints: 1051
    Root location: /content/data/train
    StandardTransform
Transform: Compose(
               Resize(size=(224, 224), interpolation=bilinear, max_size=None, antialias=None)
               ToTensor()
           )
test_data:  Dataset ImageFolder
    Number of datapoints: 1051
    Root location: /content/data/train
    StandardTransform
Transform: Compose(
               Resize(size=(224, 224), interpolation=bilinear, max_size=None, antialias=None)
               ToTensor()
           )
num_train:  1051
['3AndBelow', 'above3']
```

FIGURE 4.3: Splitting training and test data

Subsequently, a pretrained model will be loaded, in this project I have utilized the ResNet50 as the pretrained model. ResNet50 is a CNN model that is 50 layers deep, where the pretrained model can be used in training the deep learning model for the project. The layers of the pretrained model are displayed as below:

```
ResNet(
  (conv1): Conv2d(3, 64, kernel_size=(7, 7), stride=(2, 2), padding=(3, 3), bias=False)
  (bn1): BatchNorm2d(64, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
  (relu): ReLU(inplace=True)
  (maxpool): MaxPool2d(kernel_size=3, stride=2, padding=1, dilation=1, ceil_mode=False)
  (layer1): Sequential(
    (0): Bottleneck(
      (conv1): Conv2d(64, 64, kernel_size=(1, 1), stride=(1, 1), bias=False)
      (bn1): BatchNorm2d(64, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
      (conv2): Conv2d(64, 64, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1), bias=False)
      (bn2): BatchNorm2d(64, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
      (conv3): Conv2d(64, 256, kernel_size=(1, 1), stride=(1, 1), bias=False)
      (bn3): BatchNorm2d(256, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
      (relu): ReLU(inplace=True)
      (downsample): Sequential(
        (0): Conv2d(64, 256, kernel_size=(1, 1), stride=(1, 1), bias=False)
        (1): BatchNorm2d(256, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
      )
    )
    (1): Bottleneck(
      (conv1): Conv2d(256, 64, kernel_size=(1, 1), stride=(1, 1), bias=False)
      (bn1): BatchNorm2d(64, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
      (conv2): Conv2d(64, 64, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1), bias=False)
      (bn2): BatchNorm2d(64, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
      (conv3): Conv2d(64, 256, kernel_size=(1, 1), stride=(1, 1), bias=False)
      (bn3): BatchNorm2d(256, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
      (relu): ReLU(inplace=True)
    )
    (2): Bottleneck(
      (conv1): Conv2d(256, 64, kernel_size=(1, 1), stride=(1, 1), bias=False)
      (bn1): BatchNorm2d(64, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
      (conv2): Conv2d(64, 64, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1), bias=False)
      (bn2): BatchNorm2d(64, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
      (conv3): Conv2d(64, 256, kernel_size=(1, 1), stride=(1, 1), bias=False)
      (bn3): BatchNorm2d(256, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
      (relu): ReLU(inplace=True)
```

FIGURE 4.4: Pretrained Model CNN Layer 1

```
      (relu): ReLU(inplace=True)
    )
  )
  (layer2): Sequential(
    (0): Bottleneck(
      (conv1): Conv2d(256, 128, kernel_size=(1, 1), stride=(1, 1), bias=False)
      (bn1): BatchNorm2d(128, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
      (conv2): Conv2d(128, 128, kernel_size=(3, 3), stride=(2, 2), padding=(1, 1), bias=False)
      (bn2): BatchNorm2d(128, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
      (conv3): Conv2d(128, 512, kernel_size=(1, 1), stride=(1, 1), bias=False)
      (bn3): BatchNorm2d(512, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
      (relu): ReLU(inplace=True)
      (downsample): Sequential(
        (0): Conv2d(256, 512, kernel_size=(1, 1), stride=(2, 2), bias=False)
        (1): BatchNorm2d(512, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
      )
    )
    (1): Bottleneck(
      (conv1): Conv2d(512, 128, kernel_size=(1, 1), stride=(1, 1), bias=False)
      (bn1): BatchNorm2d(128, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
      (conv2): Conv2d(128, 128, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1), bias=False)
      (bn2): BatchNorm2d(128, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
      (conv3): Conv2d(128, 512, kernel_size=(1, 1), stride=(1, 1), bias=False)
      (bn3): BatchNorm2d(512, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
      (relu): ReLU(inplace=True)
    )
    (2): Bottleneck(
      (conv1): Conv2d(512, 128, kernel_size=(1, 1), stride=(1, 1), bias=False)
      (bn1): BatchNorm2d(128, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
      (conv2): Conv2d(128, 128, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1), bias=False)
      (bn2): BatchNorm2d(128, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
      (conv3): Conv2d(128, 512, kernel_size=(1, 1), stride=(1, 1), bias=False)
      (bn3): BatchNorm2d(512, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
      (relu): ReLU(inplace=True)
    )
    (3): Bottleneck(
      (conv1): Conv2d(512, 128, kernel_size=(1, 1), stride=(1, 1), bias=False)
```

FIGURE 4.5: Pretrained Model CNN Layer 2

```
      (relu): ReLU(inplace=True)
    )
  )
  (layer3): Sequential(
    (0): Bottleneck(
      (conv1): Conv2d(512, 256, kernel_size=(1, 1), stride=(1, 1), bias=False)
      (bn1): BatchNorm2d(256, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
      (conv2): Conv2d(256, 256, kernel_size=(3, 3), stride=(2, 2), padding=(1, 1), bias=False)
      (bn2): BatchNorm2d(256, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
      (conv3): Conv2d(256, 1024, kernel_size=(1, 1), stride=(1, 1), bias=False)
      (bn3): BatchNorm2d(1024, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
      (relu): ReLU(inplace=True)
      (downsample): Sequential(
        (0): Conv2d(512, 1024, kernel_size=(1, 1), stride=(2, 2), bias=False)
        (1): BatchNorm2d(1024, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
      )
    )
    (1): Bottleneck(
      (conv1): Conv2d(1024, 256, kernel_size=(1, 1), stride=(1, 1), bias=False)
      (bn1): BatchNorm2d(256, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
      (conv2): Conv2d(256, 256, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1), bias=False)
      (bn2): BatchNorm2d(256, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
      (conv3): Conv2d(256, 1024, kernel_size=(1, 1), stride=(1, 1), bias=False)
      (bn3): BatchNorm2d(1024, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
      (relu): ReLU(inplace=True)
    )
    (2): Bottleneck(
      (conv1): Conv2d(1024, 256, kernel_size=(1, 1), stride=(1, 1), bias=False)
      (bn1): BatchNorm2d(256, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
      (conv2): Conv2d(256, 256, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1), bias=False)
      (bn2): BatchNorm2d(256, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
      (conv3): Conv2d(256, 1024, kernel_size=(1, 1), stride=(1, 1), bias=False)
      (bn3): BatchNorm2d(1024, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
      (relu): ReLU(inplace=True)
    )
    (3): Bottleneck(
      (conv1): Conv2d(1024, 256, kernel_size=(1, 1), stride=(1, 1), bias=False)
```

FIGURE 4.6: Pretrained Model CNN Layer 3

```
  )
  (layer4): Sequential(
    (0): Bottleneck(
      (conv1): Conv2d(1024, 512, kernel_size=(1, 1), stride=(1, 1), bias=False)
      (bn1): BatchNorm2d(512, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
      (conv2): Conv2d(512, 512, kernel_size=(3, 3), stride=(2, 2), padding=(1, 1), bias=False)
      (bn2): BatchNorm2d(512, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
      (conv3): Conv2d(512, 2048, kernel_size=(1, 1), stride=(1, 1), bias=False)
      (bn3): BatchNorm2d(2048, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
      (relu): ReLU(inplace=True)
      (downsample): Sequential(
        (0): Conv2d(1024, 2048, kernel_size=(1, 1), stride=(2, 2), bias=False)
        (1): BatchNorm2d(2048, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
      )
    )
    (1): Bottleneck(
      (conv1): Conv2d(2048, 512, kernel_size=(1, 1), stride=(1, 1), bias=False)
      (bn1): BatchNorm2d(512, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
      (conv2): Conv2d(512, 512, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1), bias=False)
      (bn2): BatchNorm2d(512, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
      (conv3): Conv2d(512, 2048, kernel_size=(1, 1), stride=(1, 1), bias=False)
      (bn3): BatchNorm2d(2048, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
      (relu): ReLU(inplace=True)
    )
    (2): Bottleneck(
      (conv1): Conv2d(2048, 512, kernel_size=(1, 1), stride=(1, 1), bias=False)
      (bn1): BatchNorm2d(512, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
      (conv2): Conv2d(512, 512, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1), bias=False)
      (bn2): BatchNorm2d(512, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
      (conv3): Conv2d(512, 2048, kernel_size=(1, 1), stride=(1, 1), bias=False)
      (bn3): BatchNorm2d(2048, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
      (relu): ReLU(inplace=True)
    )
  )
  (avgpool): AdaptiveAvgPool2d(output_size=(1, 1))
  (fc): Linear(in_features=2048, out_features=1000, bias=True)
```

FIGURE 4.7: Pretrained Model CNN Layer 4

Consequently, a loss function is then created, along with an optimizer and learning rate. Loss functions are critical components of neural networks and is used to evaluate a candidate solution. Loss is nothing more than a Neural Net prediction mistake. And the procedure for calculating the loss is known as the Loss Function.

```
[46] for param in model.parameters():
         param.requires_grad = False

     model.fc = nn.Sequential(nn.Linear(2048, 512),
                                          nn.ReLU(),
                                          nn.Dropout(0.2),
                                          nn.Linear(512, 10),
                                          nn.LogSoftmax(dim=1))
     criterion = nn.NLLLoss()
     optimizer = optim.Adam(model.fc.parameters(), lr=0.003)
     model.to(device)
```

FIGURE 4.8: Creation of loss function and optimizer and learning rate

Afterwards, the deep learning training model trained on a dataset of 1051 children's drawings is done. Firstly, the batches of children's drawings are loaded and done in a forward loop, which will then calculate the loss function. Subsequently the Adam optimizer will be used. The losses are displayed as below, and the accuracy is calculating for every batch of epochs. In this project, I used 5 epochs values as the number of batches to train the deep learning model. Previously, I have tried to train the model with more than 5 batches, and it creates a situation called "overtraining", where the model would be too sensitive to every temporal arches and strokes of the images. Therefore, I use the epochs with only 5 batches, resulting in the end model accuracy of 0.896, equivalent to 89.6% of accuracy.

```
epochs = 5
steps = 0
running_loss = 0
print_every = 10
train_losses, test_losses = [], []
for epoch in range(epochs):
    for inputs, labels in trainloader:
        steps += 1
        inputs, labels = inputs.to(device), labels.to(device)
        optimizer.zero_grad()
        logps = model.forward(inputs)
        loss = criterion(logps, labels)
        loss.backward()
        optimizer.step()
        running_loss += loss.item()

        if steps % print_every == 0:
            test_loss = 0
            accuracy = 0
            model.eval()
            with torch.no_grad():
                for inputs, labels in testloader:
                    inputs, labels = inputs.to(device),labels.to(device)
                    logps = model.forward(inputs)
                    batch_loss = criterion(logps, labels)
                    test_loss += batch_loss.item()

                    ps = torch.exp(logps)
                    top_p, top_class = ps.topk(1, dim=1)
                    equals = top_class == labels.view(*top_class.shape)
                    accuracy +=torch.mean(equals.type(torch.FloatTensor)).item()
            train_losses.append(running_loss/len(trainloader))
            test_losses.append(test_loss/len(testloader))
            print(f"Epoch {epoch+1}/{epochs}.. "
                  f"Train loss: {running_loss/print_every:.3f}.. "
                  f"Test loss: {test_loss/len(testloader):.3f}.. "
                  f"Test accuracy: {accuracy/len(testloader):.3f}")
            running_loss = 0
            model.train()
torch.save(model, 'aerialmodel.pth')
```

FIGURE 4.9: Deep Learning Training Model

```
Epoch 1/5.. Train loss: 0.121.. Test loss: 0.198.. Test accuracy: 0.912
Epoch 2/5.. Train loss: 0.270.. Test loss: 0.223.. Test accuracy: 0.900
Epoch 3/5.. Train loss: 0.268.. Test loss: 0.280.. Test accuracy: 0.857
Epoch 3/5.. Train loss: 0.151.. Test loss: 0.258.. Test accuracy: 0.894
Epoch 4/5.. Train loss: 0.197.. Test loss: 0.345.. Test accuracy: 0.845
Epoch 5/5.. Train loss: 0.152.. Test loss: 0.254.. Test accuracy: 0.898
Epoch 5/5.. Train loss: 0.156.. Test loss: 0.247.. Test accuracy: 0.896
```

FIGURE 4.10: Model Accuracy Results after Training

Accordingly, the plot in Figure 4.11 below shows that with 5 epochs, the validation loss increases and then decreases towards the end. This means that after the 5th epoch, the validation loss is low, which is good in order to achieve a high accuracy with efficient sensitivity in detecting the age of the child based on their drawing as The training loss, as expected, is very low, which is good as it prevents any errors and outputs a good result for prediction.

```
plt.plot(train_losses, label='Training loss')
plt.plot(test_losses, label='Validation loss')
plt.legend(frameon=False)
plt.show()
```

FIGURE 4.11: Training and validation losses plot

4.3 The existing sketch recognition techniques are considerably static

This problem statement is addressed by the objective of to develop a sketch recognition tool engine which automates the process of identifying children's psychological development growth with CNN. With CNN as the created deep learning model's layers, the prediction of the sketch recognition is done as Figure 4.12. Consequently, the get_random_images() function in Figure 4.2 is created to pick a number of random orders of drawing images from the dataset folders to test the prediction model. Finally, the prediction function's result is predicted and printed after getting the random image sample.

```
def predict_image(image):
    image_tensor = test_transforms(image).float()
    image_tensor = image_tensor.unsqueeze_(0)
    print(image_tensor.shape)
    input = Variable(image_tensor)
    input = input.to(device)
    output = model(input)
    index = output.data.cpu().numpy().argmax()
    return index
```

```
def get_random_images(num):
    data = datasets.ImageFolder(data_dir, transform=test_transforms)
    classes = data.classes
    indices = list(range(len(data)))
    np.random.shuffle(indices)
    idx = indices[:num]
    from torch.utils.data.sampler import SubsetRandomSampler
    sampler = SubsetRandomSampler(idx)
    loader = torch.utils.data.DataLoader(data,
                   sampler=sampler, batch_size=num)
    dataiter = iter(loader)
    images, labels = dataiter.next()
    return images, labels, classes
```

```
[44] to_pil = transforms.ToPILImage()
     images, labels, classes = get_random_images(5)
     fig=plt.figure(figsize=(10,10))
     for ii in range(len(images)):
         image = to_pil(images[ii])
         index = predict_image(image)
         sub = fig.add_subplot(1, len(images), ii+1)
         res = int(labels[ii]) == index
         sub.set_title(str(classes[index]) + ":" + str(res))
         plt.axis('off')
         plt.imshow(image)
     plt.show()
```

FIGURE 4.12: Predicting the results for the recognizer

The end results below shows the predicted output, where the core engine of the sketch recognition tool outputs the predicted age categorization with ‘above3:True”, ‘3AndBelow:True’, ‘above3:False’ or ‘3AndBelow:False”. These results are based on the ability of the CNN layers to identify each strokes and train itself to identify which class it belongs to. If the predicted output image results to ‘above3: True’ or ‘3AndBelow:False”, while the child’s age is above 3 years old, meaning there are currently no developmental delays detected. However, if the predicted output image results to ‘above3:False’ and ‘3AndBelow:True’ while the child’s age is above 3 years old, there are existing developmental delays and possible psychological disorders prevalent, where it is best to bring the child for further check-up.

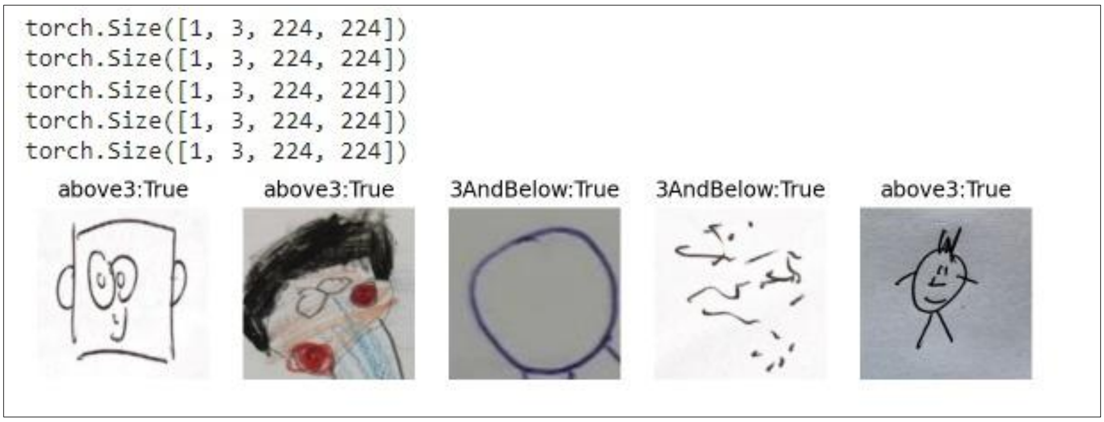

FIGURE 4.13: The end result of the Sketch Recognition based on Deep Learning for Children’s Psychological Development Monitoring project.

CHAPTER 5

CONCLUSION AND RECOMMENDATIONS

5. CONCLUSION AND RECOMMENDATIONS

The basis of a sketch recognition tool focuses on children's drawing can open up many possibilities' opportunity in future. It can be a tool to understand children's psychological developmental growth by extracting visual processing information from the temporal and order of their sketches. It can also assist as a visual resource for children's literacy processes. In particular, it can help challenging children to learn and write better via visually mediated task with this tool. For example, children with dyslexia can think primarily in pictures better than words, thus the tool can use to support their learning strategy. The tool can also be extended to analyse children's state and emotional well-being where children's psychological mood can be portrayed through literal (direct) and non-literal (indirect) drawings. In suffice, the development of a sketch recognition tool for children can give numerous contributions to many areas.

As this project is an age-specific identification that is built on existing Deep Learning technologies, especially Convolutional Neural Network (CNN), the resultant tool is intended to open the way for a huge amount of study in children's psychological development in the future, such as virtually detecting emotions through their drawings. Further recommendations on the software side of the research project, is by providing a platform for the core engine of this project to run on. For example, a web application interface that would display the overall functions and print out the results for the users to see. As this project has reached its completion and achieved its stated objectives by addressing the problem statements by using the RAD methodology, deep learning has proven to bring a powerful impact in terms of its deep neural network which offers high accuracy and positive impact to the targeted users.

REFERENCES

Abell, S. C., Von Briesen, P. D., & Watz, L. S. (1996). Intellectual evaluations of children using human figure drawings: An empirical investigation of two methods. *Journal of Clinical Psychology*, *52*(1), 67–74. https://doi.org/10.1002/(sici)1097-4679(199601)52:1<67::aid-jclp9>3.0.co;2-t.

Choo, Y. Y., Agarwal, P., How, C. H., & Yeleswarapu, S. P. (2019). Developmental delay: identification and management at primary care level. *Singapore Medical Journal*, *60*(3), 119–123. https://doi.org/10.11622/smedj.2019025.

Hijazi, S., Kumar, R., & Rowen, C. (2015). Using Convolutional Neural Networks for Image Recognition. *Cadence*, 1-3.

Kim, H., Taele, P., Seo, J., Liew, J., & Hammond, T. (2016). EasySketch2: A Novel Sketch- based Interface for Improving Children's Fine Motor Skills and School Readiness. *Eurographics DL Home.* https://diglib.eg.org/handle/10.2312/exp20161065.

Laak, J. T., De Goede, M., Aleva, A., & Rijswijk, P. V. (2005). The Draw-A-Person Test: An Indicator of Children's Cognitive and Socioemotional Adaptation? The Journal of Genetic Psychology, 166(1), 77–93. https://doi.org/10.3200/gntp.166.1.77-93.

Leibowitz, M. (1999). Interpreting projective drawings: A self-psychological approach. *Psychology Press*. https://doi.org/10.4324/9780203777992.

Li, Q., Cai, W., Wang, X., Zhou, Y., Feng, D. D., & Chen, M. (2014). Medical image classification with convolutional neural network. 2014 13th International Conference on Control Automation Robotics & Vision (ICARCV). doi:10.1109/icarcv.2014.7064414.

Malchiodi, C. A. (1998). A Historical Perspective on Children's Drawings. In *Understanding children's drawings*, 1–9.

Meiyin Wu, & Li Chen. (2015). Image recognition based on deep learning. *2015 Chinese Automation Congress (CAC)*. https://doi.org/10.1109/cac.2015.7382560.

Nezafat Maldonado, B., Chandna, J., & Gladstone, M. (2019). A systematic review of tools used to screen and assess for externalising behaviour symptoms in low and middle income settings. *Global Mental Health*, *6*. https://doi.org/10.1017/gmh.2019.11.

Sriram, R. (2020). *Why ages 2-7 matter so much for brain development*. Edutopia. Retrieved November 29, 2021, from https://www.edutopia.org/article/why-ages-2-7-matter-so-much-brain-development.

Strommen, E. (1988). A century of children drawing: The evolution of theory and research concerning the drawings of children. *Visual Arts Research*, 14(2)[28], 13–24.

Xu, P., Joshi, C. K., & Bresson, X. (2019). Multigraph Transformer for Free-Hand Sketch Recognition. IEEE Transactions on Neural Networks and Learning Systems, 1–12. https://doi.org/10.1109/tnnls.2021.3069230.