Artificial Intelligence (AI)
Final Project Presentation

Class: IBM IMVAI-2202

Team: 1

Team Members: Amy Poh Bee Whee

Jennifer Mo Li Hua

Soo Weng Wah

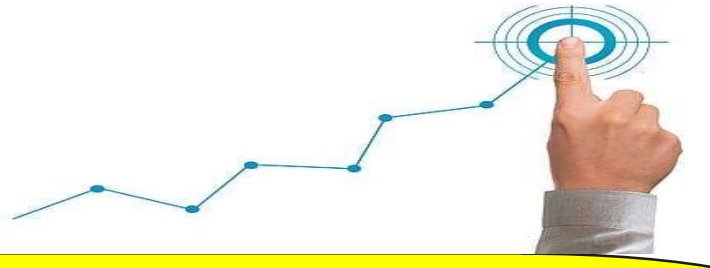
Teo Kim Keong

Teo Teng Teng

AGENDA

Teng Teng	• Project Objective & Topic • Project Planning & Milestones • Enterprise Design Thinking
Jennifer	• AI Essentials Framework • Model Overview
Weng Wah	• Chatbot Design
Amy	• Implementation Overview • Live Demo
Kim Keong	• Integration & Deployment • Project Learning

PROJECT OBJECTIVE



Demonstrate an AI solution addressing a specific business challenge

TOOLS

Watson Assistant
(Chatbot)

Mural
(collaboration and Enterprise Design Thinking and AI Design Essentials)

Flask
(Integration)

Heroku
Web deployment

GitHub
Repository purpose

Kaggle
(Jupyter Notebooks environment)

Brackets
(Code editor)

HEROKU

kaggle

Brackets
open-source code editor
built with the web for the web

PROJECT TOPIC: AI in Early Age Education

Preparing our future generation for the Web 3.0 while building a strong foundation in their fundamentals arithmetic literacy

Introducing AI to children at a young age helps them become open-minded curious adults who are always on the brink of innovation. Their inventions might just change the world someday.

Allow children to develop problem-solving skills and logical analysis & aptitude at a very young age which helps them reach their full potential earlier

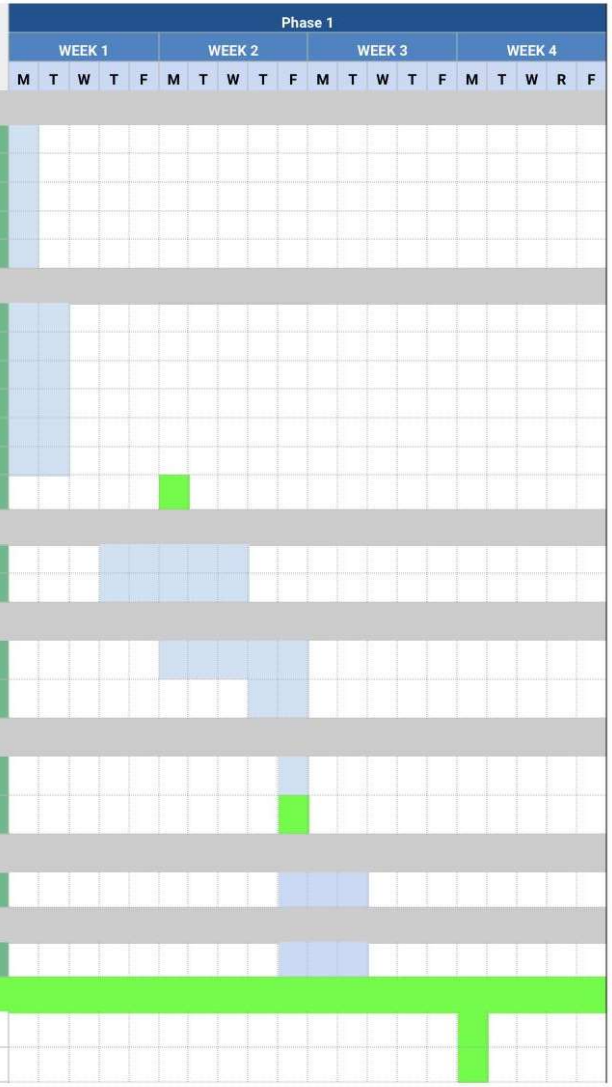
AI can positively impact on young children's learning and development

Project Planning & Milestones

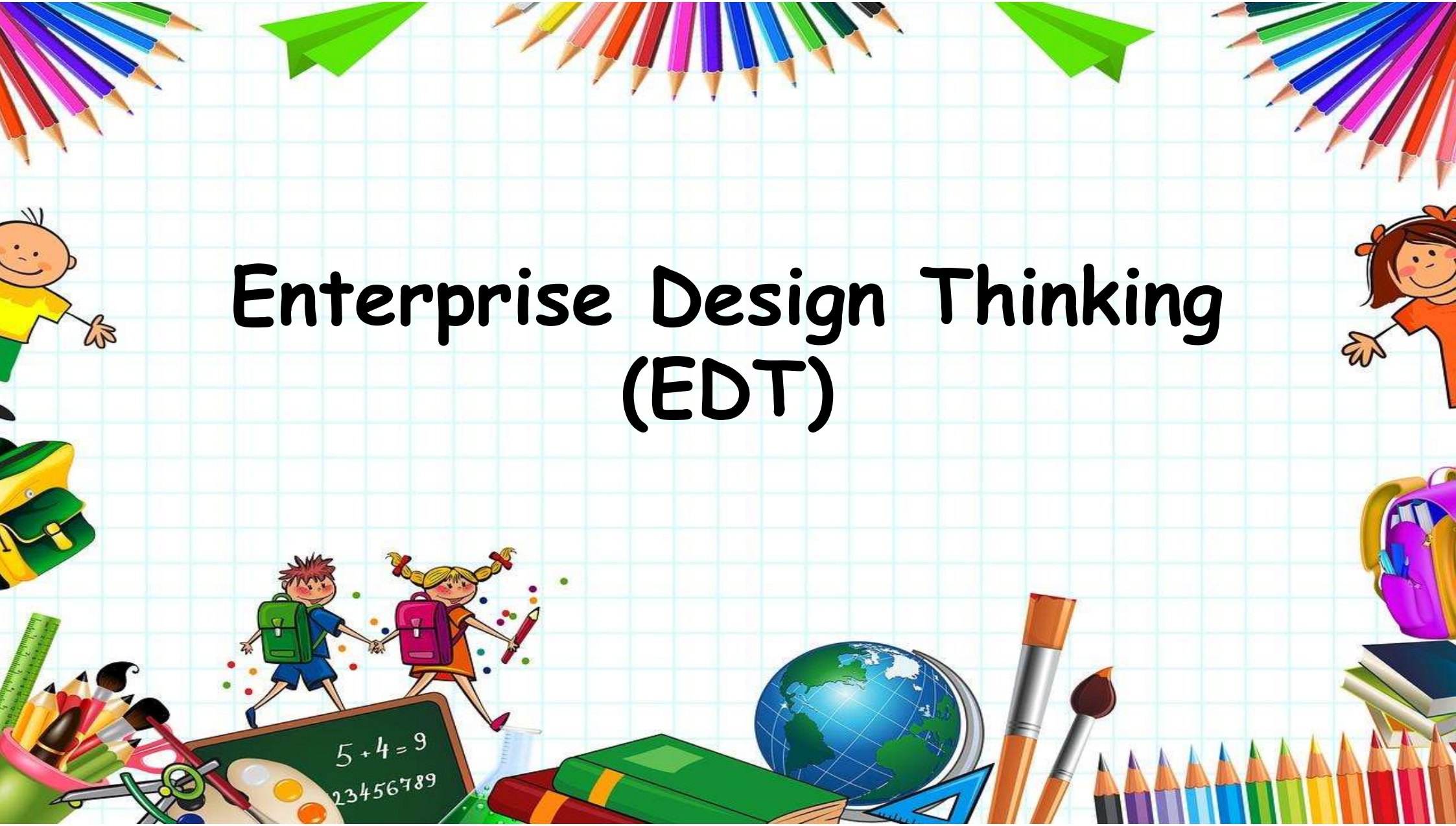
PROJECT TIMELINE

GANTT CHART

WBS NUMBER	TASK TITLE	TASK OWNER	START DATE	DUE DATE	DURATION	PCT OF TASK COMPLETE
1	**Project Definition and Planning**					
1.1	Enterprise Design Thinking	Group	22/08/22	22/08/22	4	100
1.2	Project Allocation	Group	22/08/22	22/08/22	3	100
1.3	AI Essential	Group	22/08/22	22/08/22	0	100
1.4	Project Planning	Group	22/08/22	22/08/22	0	100
1.4	Project Design	Group	22/08/22	22/08/22	0	100
2	**Model Building**					
2.1	Research & Data Sourcing	Jennifer	22/08/22	23/08/22	0	100
2.2	Data Cleansing	Jennifer	22/08/22	23/08/22	0	100
2.3	Coding	Jennifer	22/08/22	23/08/22	0	100
2.4	Model Training	Jennifer	22/08/22	23/08/22	0	100
2.5	Model Testing	Jennifer	22/08/22	23/08/22	0	100
2.6	Integration & Deployment	Jennifer	22/08/22	23/08/22	0	100
3	**Midpoint Discussion 1**	Group	29/08/22	29/08/22	0	100
4	**Chatbot**					
3.1	Chatbot Conversation Flow Design	TT, KK, Weng, Amy	25/08/22	31/08/22	0	100
3.2	Chatbot Building & Embedding	TT, KK, Weng, Amy	26/08/22	31/08/22	0	100
5	**Website HTML Coding**					
5.1	HTML Coding	Amy	29/08/22	02/09/22	0	100
5.2	HTML Integration & Testing	Amy	01/09/22	02/09/22	0	100
6	**Integration & Deployment**					
6.1	Flask Deployment	Amy	2/9/22	02/09/22	0	100
7	**Midpoint Discussion 2**	Group	2/9/22	02/09/22	0	100
8	**Project Learnings**					
8.1	Constraints & Challenges	KK	02/09/22	06/09/22	0	100
9	**Presentation Slides**					
9.1	Presentation Slides Touch Up	TT	02/09/22	09/09/22	0	100
10	**Final Presentation**		**12/09/22**	**12/09/22**		
10.1	Presentation	Group	12/09/22	12/09/22	0	0%
10.2	Presentation Submission	Group	12/09/22	12/09/22	0	0%



Enterprise Design Thinking (EDT)



PROBLEM STATEMENT

HOW MIGHT WE

use AI to guess the number handwriting of toddler

FOR parents

SO THAT

toddlers can develop legible penmanship in a fun way

PERSONA

Parents of Toddlers:

Goals

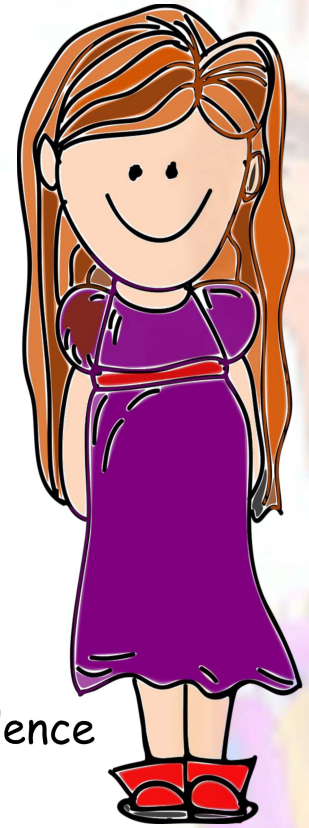
- Ensure toddlers can write numbers legibly before primary school

Motivations

- To have early start in the numeracy education
- To avoid conflicts with toddler in the learning process
- To develop competence in number handwriting thus encouraging child self confidence

Needs

- An interactive tool to teach and guide numeric handwriting

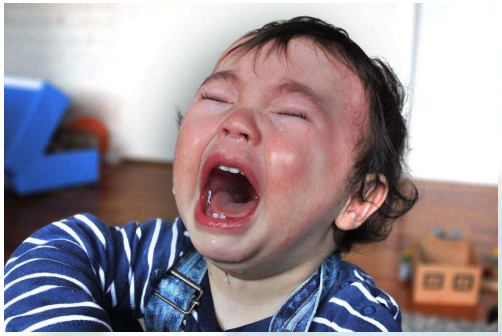


Elizabeth, 30 years' old
Mother of a 2 year old toddler
Working Full Time, Tech Savvy

PAIN POINTS

1. **Challenging to create happy learning environment for children**
   - Not to be the bad guy when correcting child's handwriting
   - Remove subjectivity when correcting child's handwriting
   - A lot of patience required when teaching kids
   - Screaming sessions when correcting child's handwriting

2. **Lack of cost effective and accessible tools better coach penmanship to toddlers**

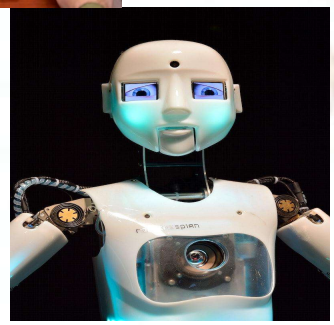


EDT - BIG IDEAS

Virtual lessons

Use **metaverse** avatar (chatbot) to teach

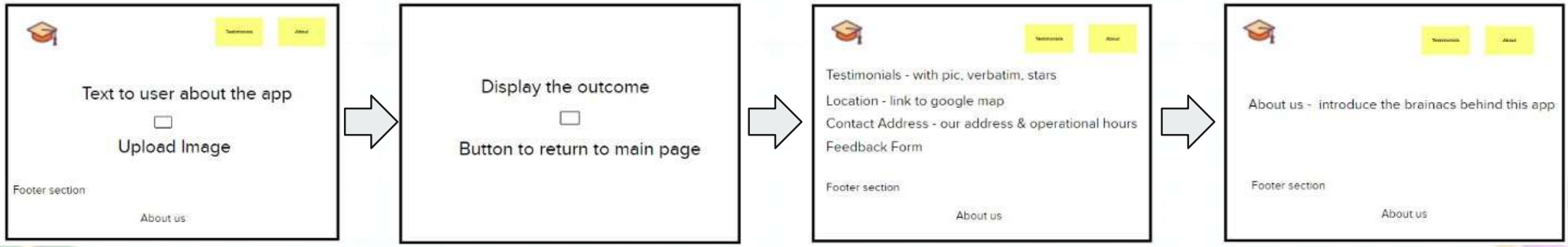
Create an interactive website where number writings are scanned for AI to recognise the image, and guess the input



Game app (writing on screen)

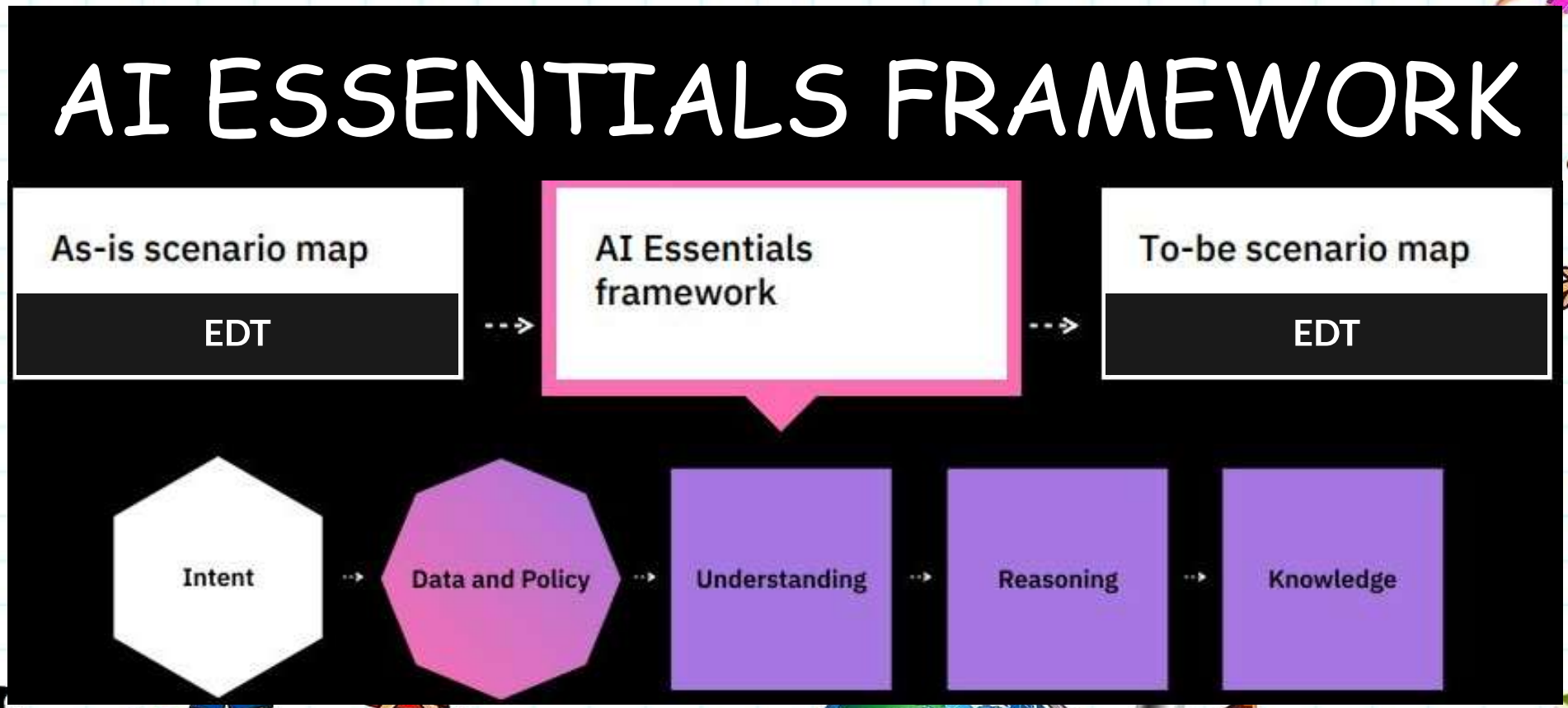
Use robot to teach children how to write

MVP: Create an interactive website where number writings are scanned for AI to recognise the image, and guess the input.



The first screen is the landing page and it is also houses the main function of this app.

AI ESSENTIALS FRAMEWORK



AI INTENT

Core AI Intent

- **Recommend with confidence**
  - Make confident guess by using machine learning and image recognition techniques to evaluate if a kid's handwriting is conforming to school teaching.
- **Enrich users' interactions**
  - Chatbot is able to respond to parents' queries with tailored dialogue and personalized experiences.

DETERMINE DATA (1/2)

PUBLIC DATA	USER DATA
HAVE - MNIST (60k images) - EMNIST Digits (240k images) - Street View House Numbers (SVHN) Dataset (600k images)	**HAVE** - N.A.
WANT / NEED - Images of handwritten digit 0 - 9	**WANT / NEED** - Handwritten digit 0 - 9 by kids on plain paper
NICE TO HAVE - Handwritten Chinese characters from 0 -9	**NICE TO HAVE** - N.A.

DETERMINE DATA (2/2)

- **EMNIST** is an extension of MNIST (Modified National Institute of Standards and Technology) to handwritten letters.

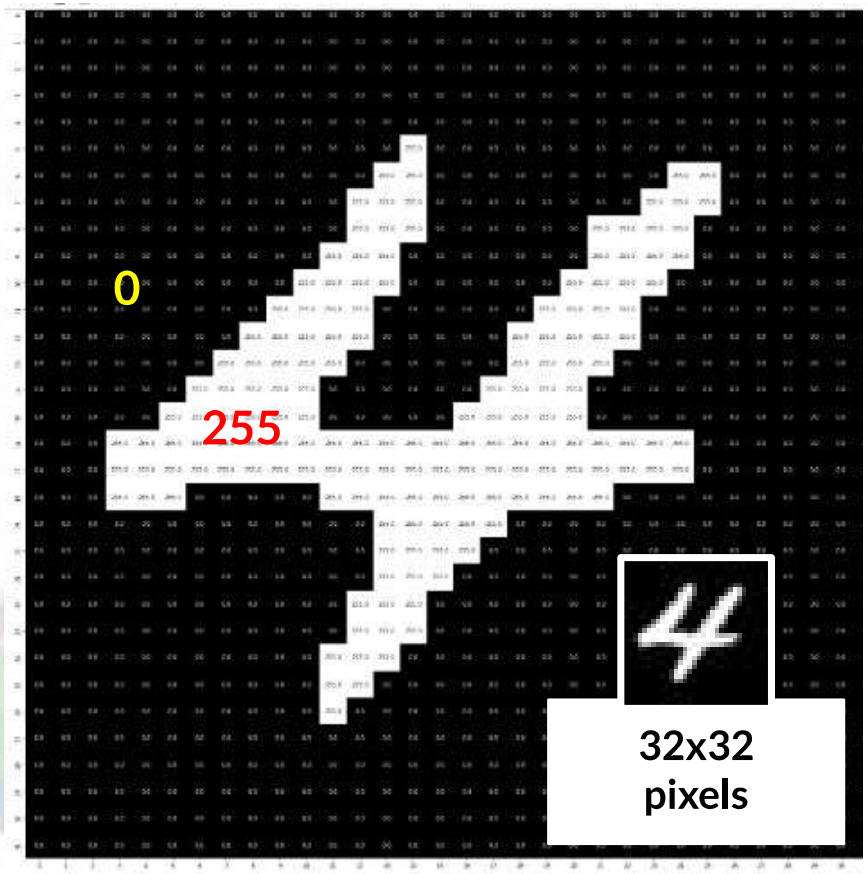
  Source: Cohen, G., Afshar, S., Tapson, J., & van Schaik, A. (2017). EMNIST: an extension of MNIST to handwritten letters. Retrieved from http://arxiv.org/abs/1702.05373

- Our dataset is downloaded from **Kaggle**: https://www.kaggle.com/datasets/vaibhao/handwritten-characters

Understanding

What will a machine need to understand from this data so it can reason?

UNDERSTANDING DATA (1/2)

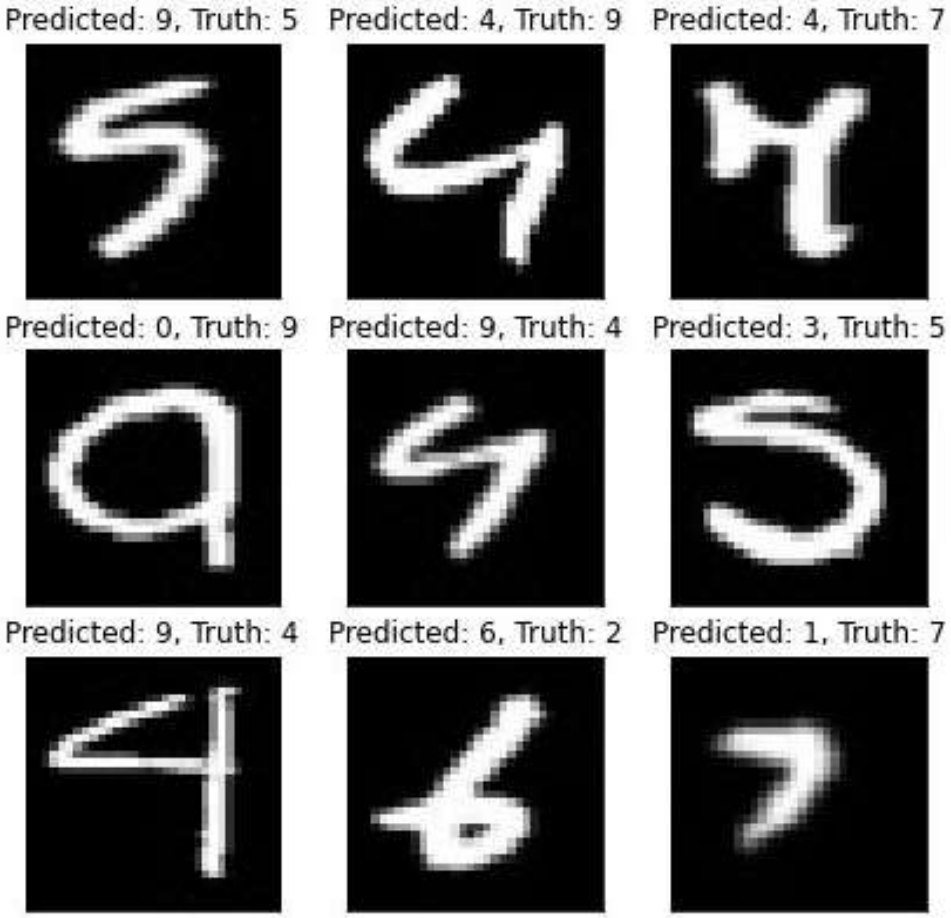


- An image is stored in a computer by representing it as an array of numbers.
- Each number indicates the brightness of a given pixel - **0** (black), **255** (white)

Understanding

What will a machine need to understand from this data so it can reason?

UNDERSTANDING DATA (2/2)



Reasoning

How will a machine apply logic to what it knows?

REASONING

- **Image recognition** is about **finding pattern** in the given image and making **prediction** with reference to a **matrix** table.
- An example of simplified "machine learning"for handwritten digit.

Pattern

	∩	⊂	∪	⊃	—	|	/	\
0	1	1	1	1	0	0	0	0
1	0	0	0	0	0	1	1	0
2	1	1	0	1	1	0	1	0
3	0	0	1	1	1	0	1	0
4	0	0	0	0	1	2	0	0
5	0	0	1	1	1	1	0	0
6	2	1	1	1	0	0	0	0
7	0	0	0	0	2	0	1	0
8	2	2	2	2	0	0	0	0
9	1	1	2	1	0	1	0	0

8

Shapes

∩	2
⊂	2
∪	2
⊃	2
/	1
|	1

Prediction

8 (8 / 10)

9 (6 / 10)

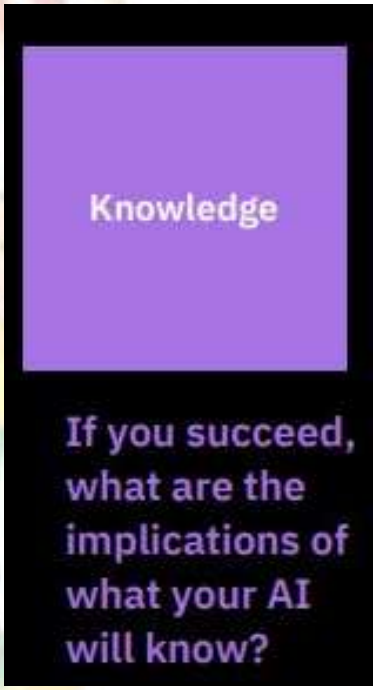
Reasoning

How will a machine apply logic to what it knows?

REASONING STATEMENT

- Our App, '**I can write**' can guess toddlers' handwritten digits by adopting power of **AI** to learn from **30K images** of handwritten digit 0 - 9 so that less subjective criteria for legibility of handwriting is being applied for all time.

- This App also helps to avoid kids throwing tantrum when parents try to correct their handwriting, hence create a happy learning atmosphere.



KNOWLEDGE

Primary effects

- Kids get to develop handwriting legibility
- Good bonding between parents and kids

Secondary effects

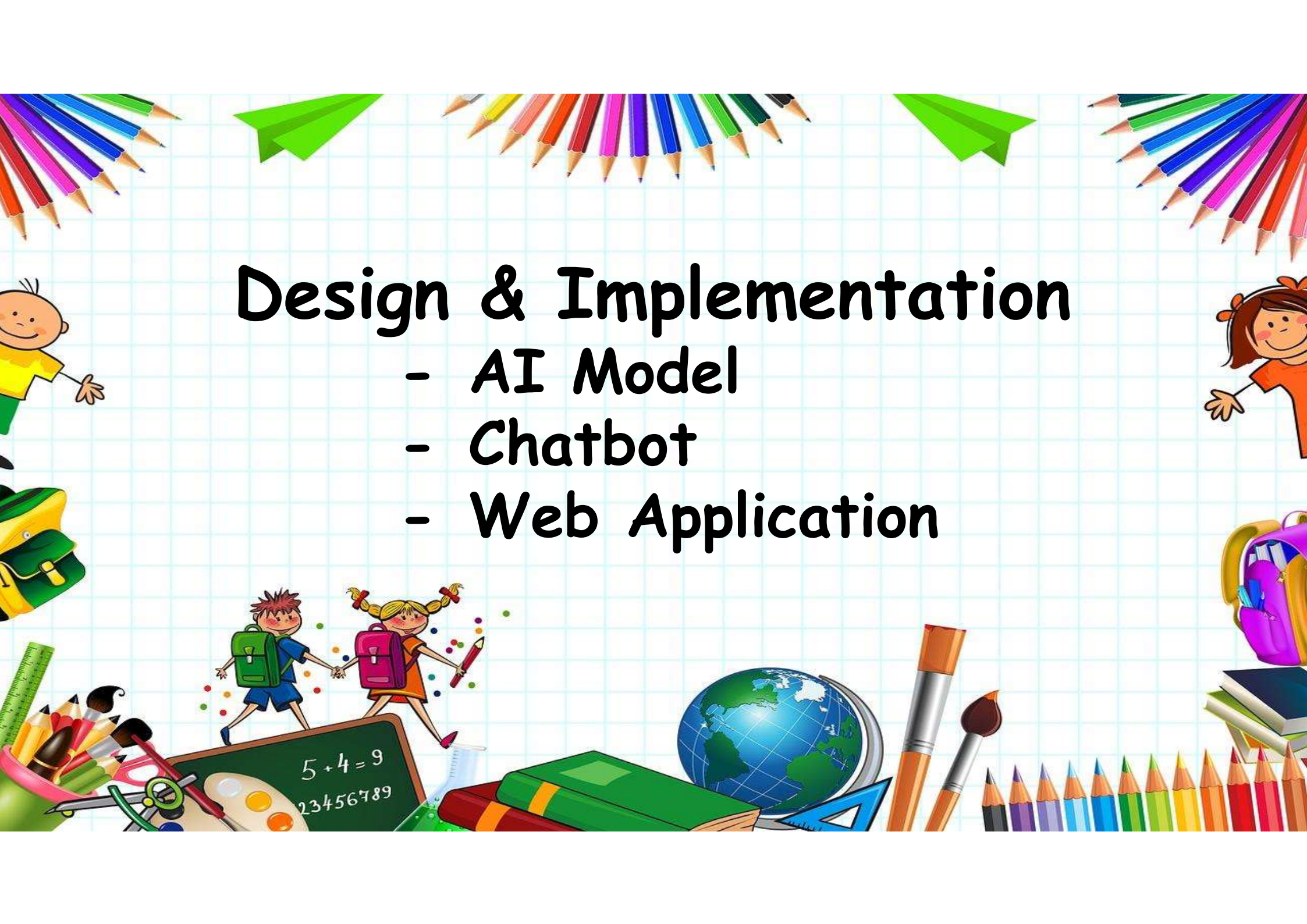
- The child gets an early exposure to AI technology
- Head start in numeracy skill

Tertiary effects

- Unintentionally cultivate "AI is always right" mindset in the child
- The child might not understand that AI model is about prediction and probability.

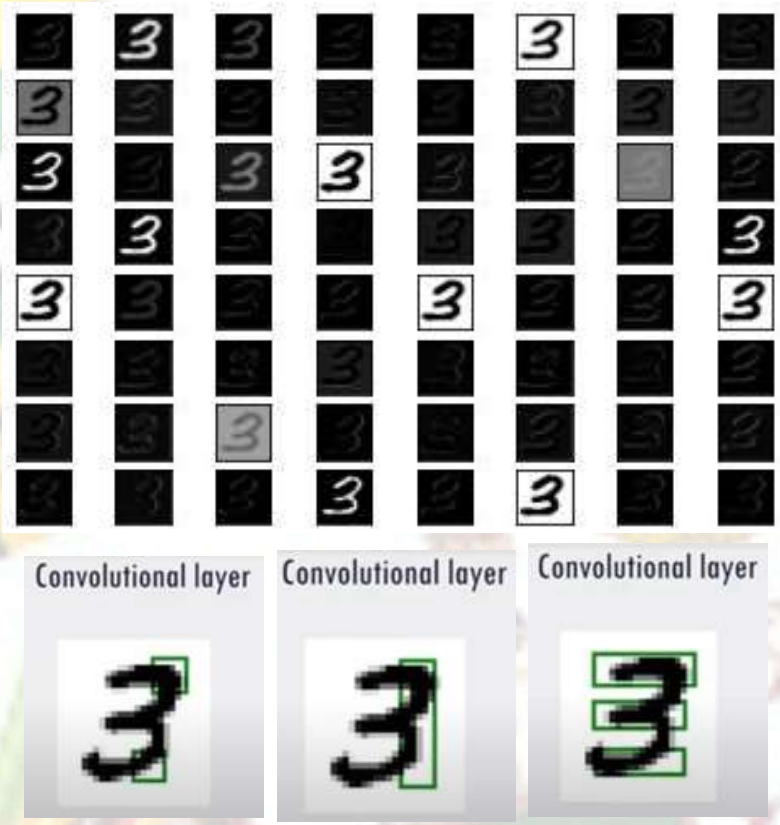
Design & Implementation

- AI Model
- Chatbot
- Web Application

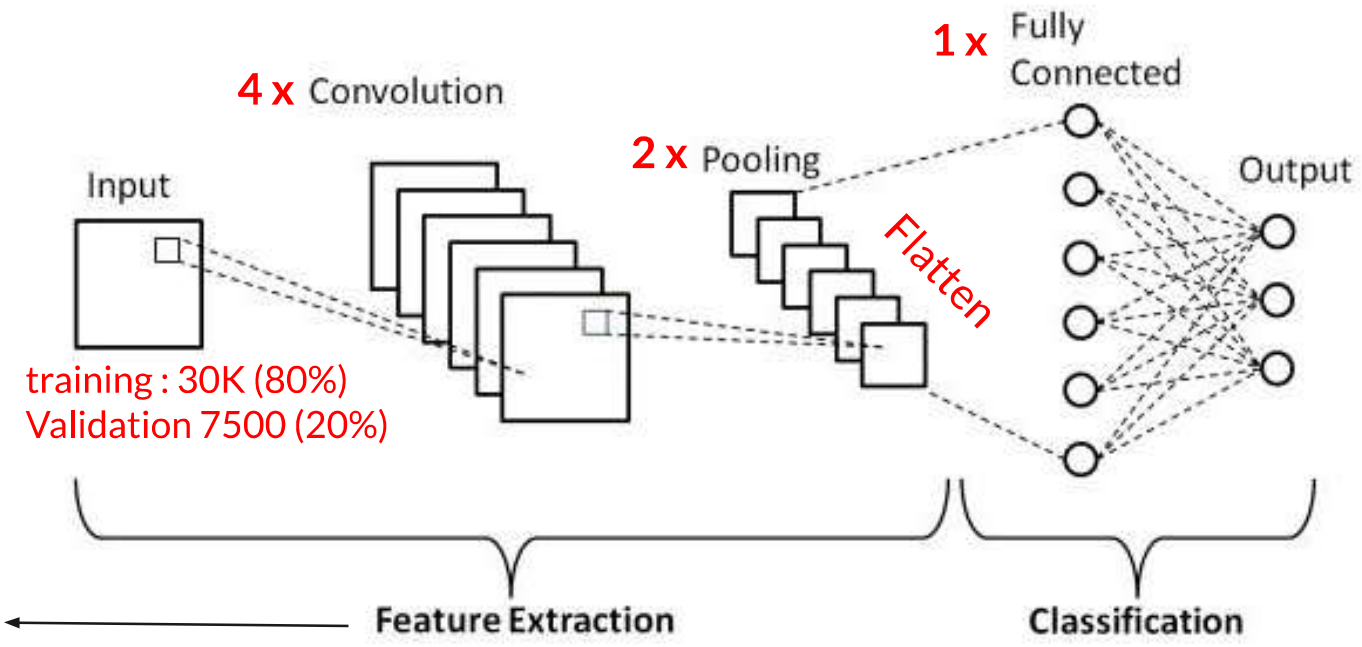


AI MODEL BUILDING using KERAS

An example of Feature Map:

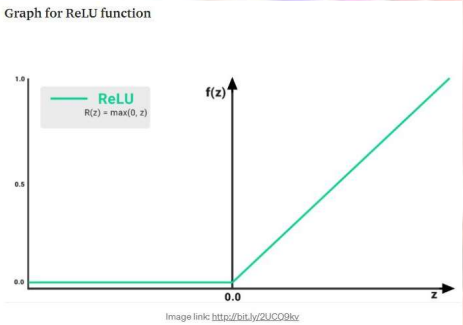


- Our AI solution is implement with **Convolutional Neural Network** (CNN) architecture.



0 1 2 3 4 5 6 7 8 9

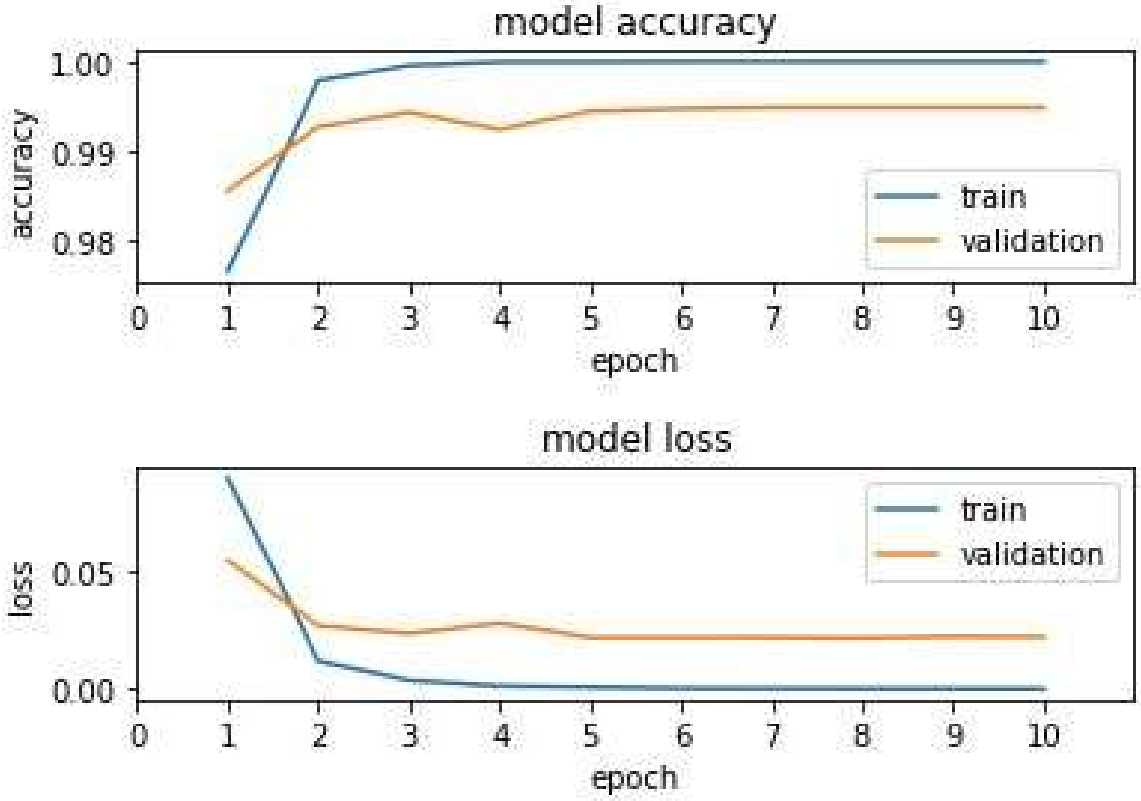
AI MODEL OPTIMISATION



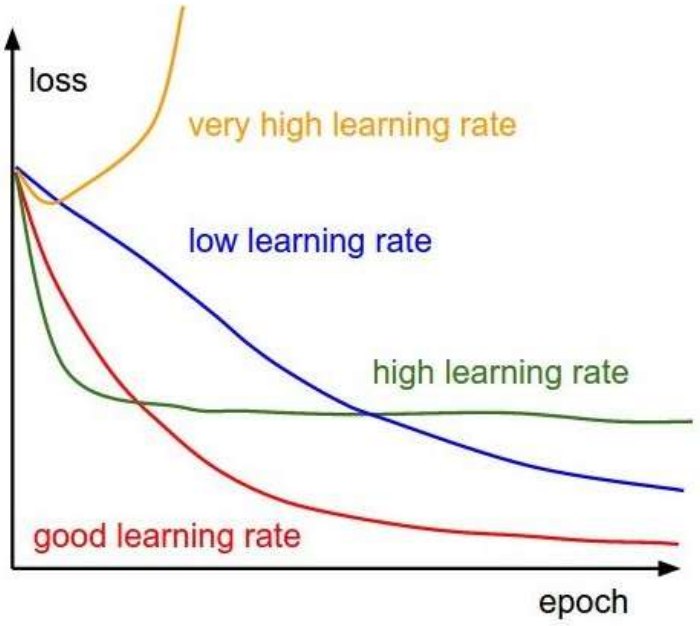
Consideration	Final Implementation	Rationale
Background color of training dataset	Keep background of training & validation **black** as it is while invert the **white** background of user data.	This is no longer a concern after we decided to convert all images from grayscale to binary image.
Choice of Keras Built-in optimizer	Adaptive optimizer: Adaptive Moment Estimation (**Adam**)	No need to focus on the learning rate value.
Neural network activation function	Rule of thumb: **Parametric ReLU (PReLu)** is another variant of ReLU and ReLU activation function goes well with **He-initialization** of weights.	PReLu aims to solve the problem of gradient's becoming zero for the left half of the axis.

AI MODEL VALIDATION

Keras



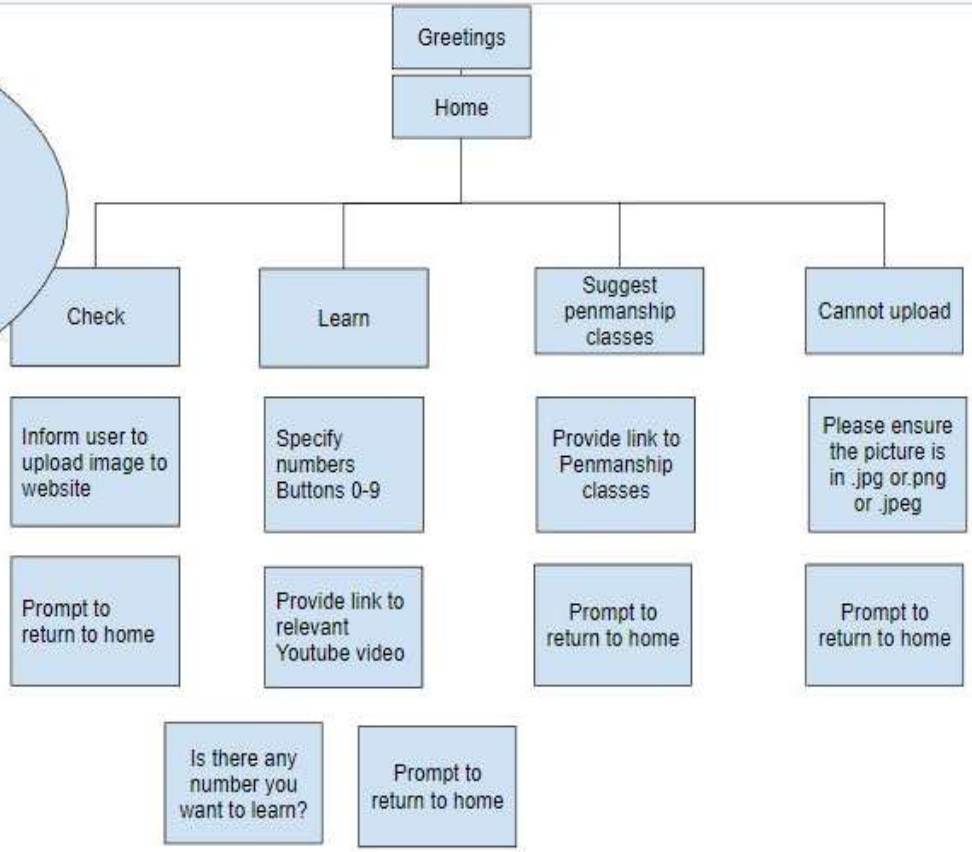
- Based on the **training loss over epoch curve**, we can tell that Adam optimizer is displaying good **learning rate**.
- It also shows that this model is not underfitting or overfitting.

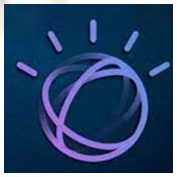


Effect of various learning rates on convergence (Img Credit: cs231n)

CHATBOT - Initial Dialog Creation

Purpose of chatbot (Leopard)
- Correct or Learn
  - Correct - scan / report outcome (iterative)
  - Learn - send link to youtube video (return control to chatbot)
- Option for online penmanship classes - online site
- Cannot upload - troubleshooting function
  -





CHATBOT - Defined Intent, Entity and Dialog Flow

Intents

1. #Guess the Number
2. #Learn how to write number
3. #Suggestion on Other Classes
4. #Help
5. #Home
6. #Greeting

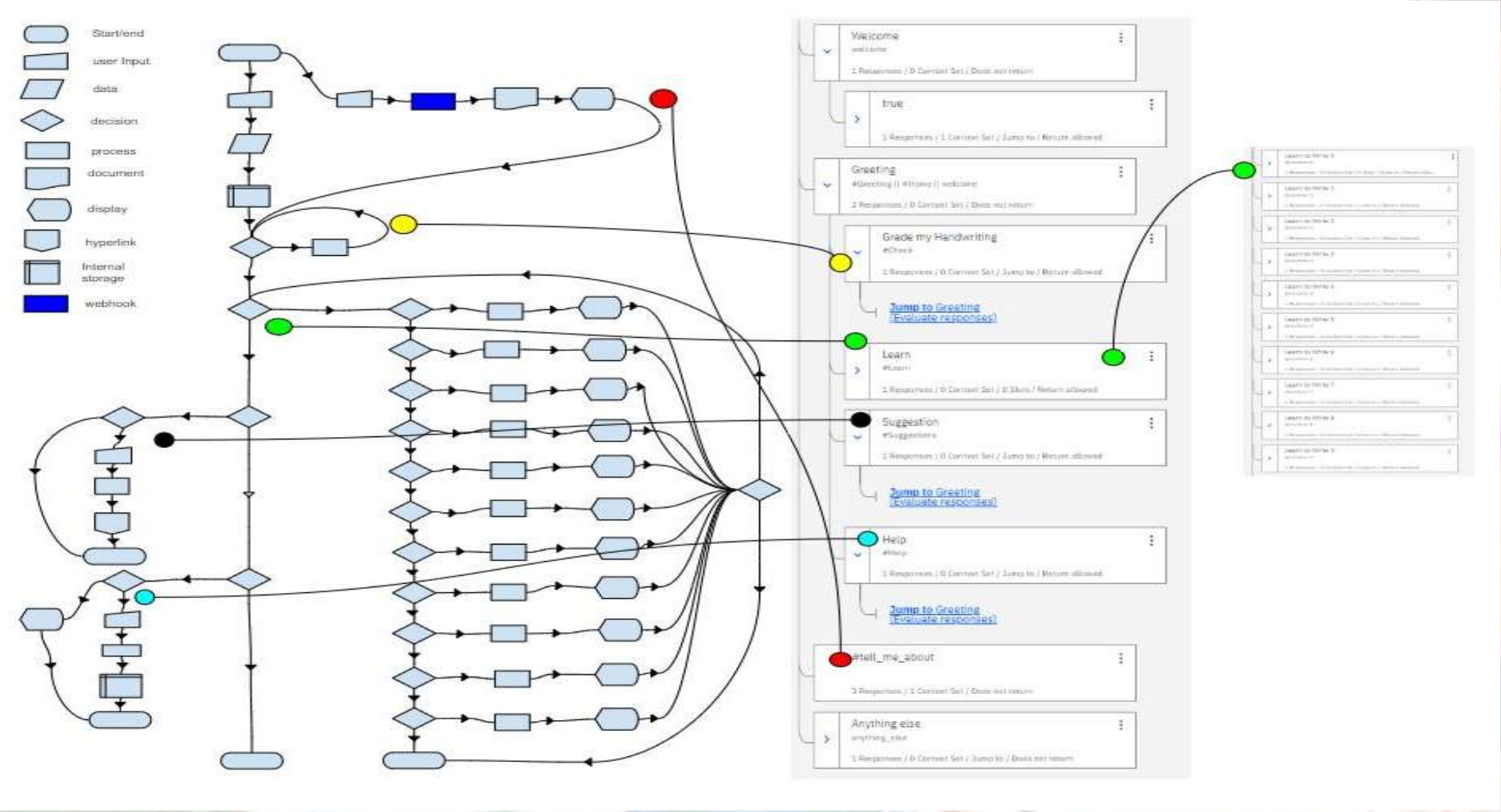
Entities

1. @Number
2. @No
3. @object_of_interest

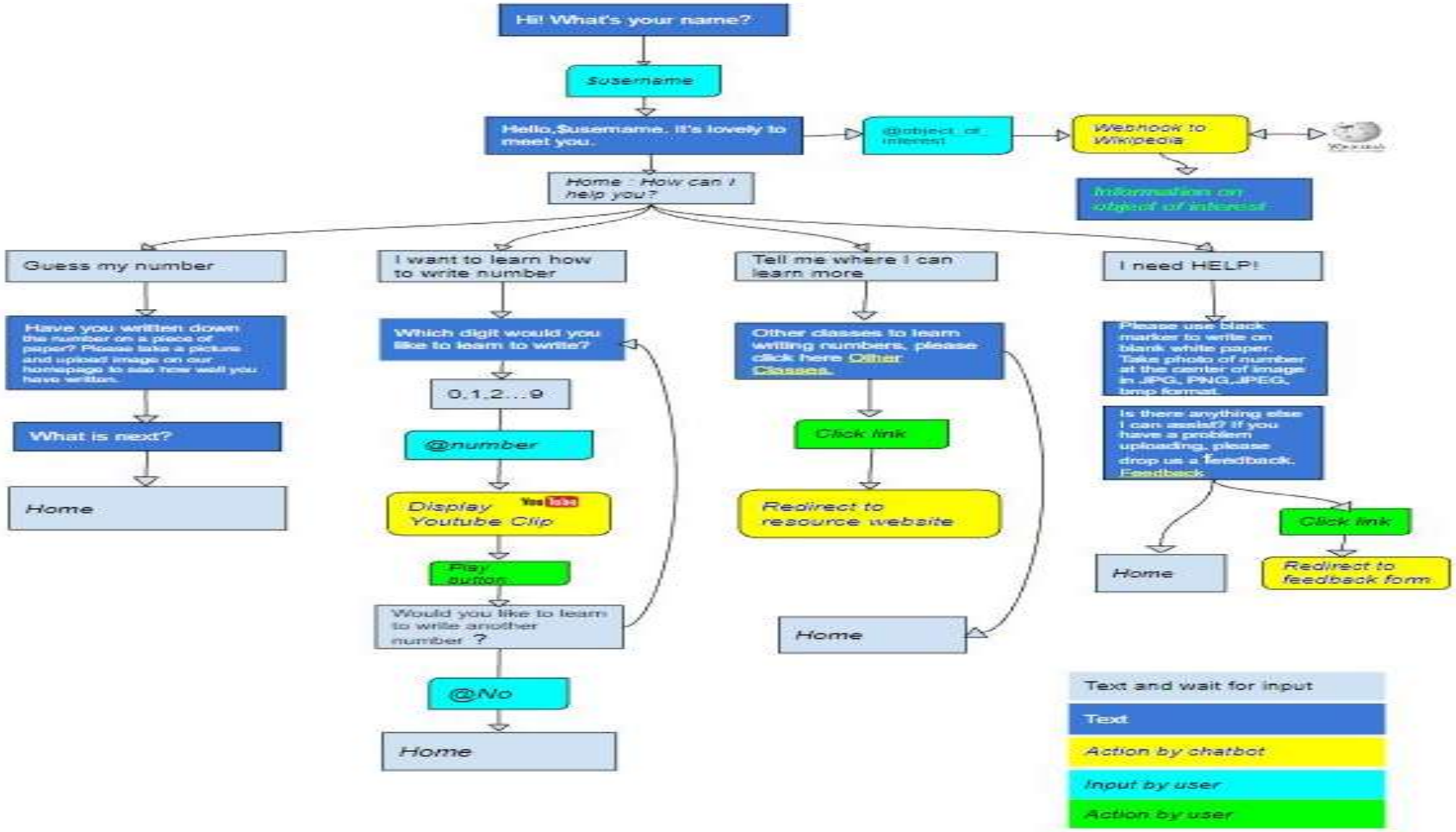
Dialogue

1. Welcome
2. Greeting
    a. Guess the Number
    b. Learn
        i. Learn to write 0
        ii. Learn to write 1
        iii. Learn to write 2
        iv. Learn to write 3
        v. Learn to write 4
        vi. Learn to write 5
        vii. Learn to write 6
        viii. Learn to write 7
        ix. Learn to write 8
        x. Learn to write 9
    c. Suggestion
    d. Help
3. Tell-me-about
4. Anything else

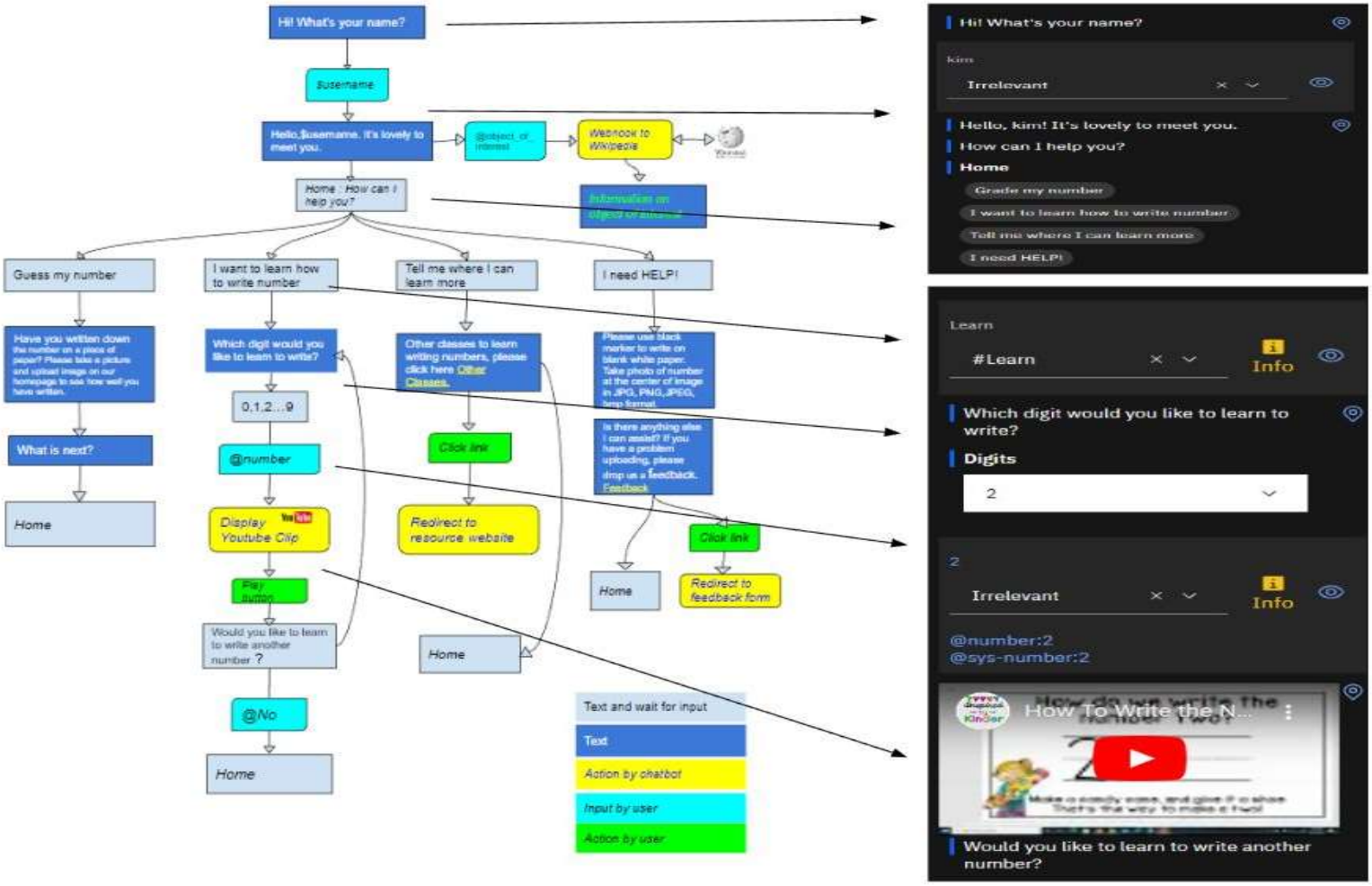
CHATBOT - Translating Flow Conversation → Dialog Nodes



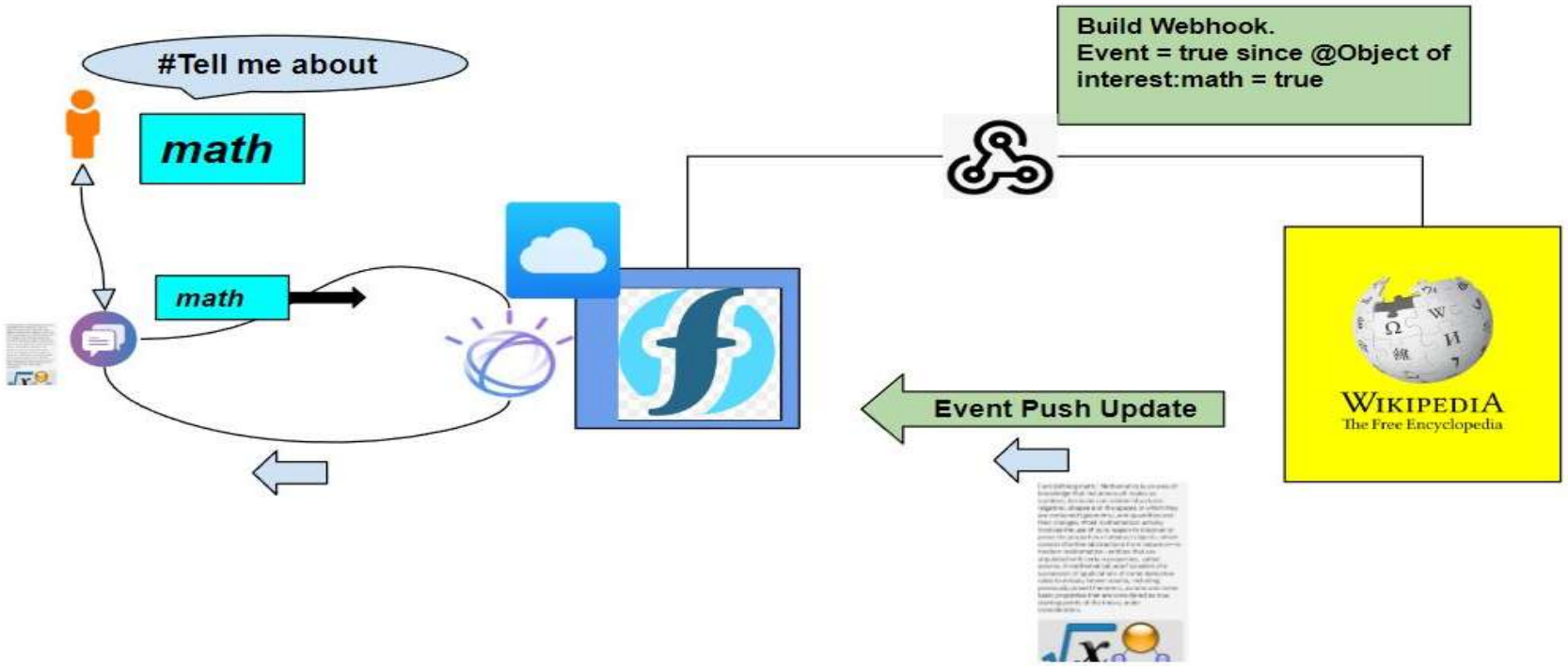
CHATBOT - Conversation Flow



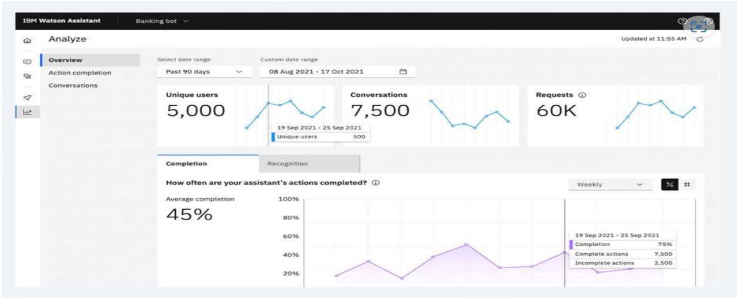
CHATBOT - Dialog Flow for Video Learning, UI/UX



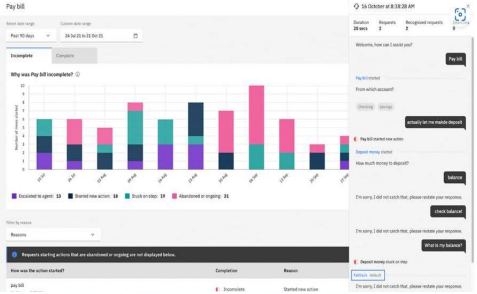
CHATBOT - Webhook Call → WIKIPEDIA



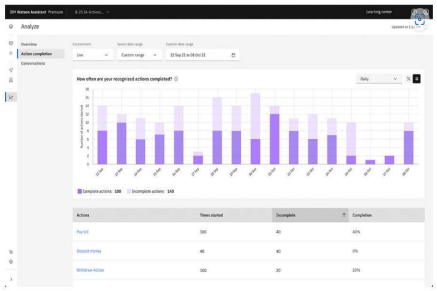
CHATBOT - Performance & Fine-Tuning



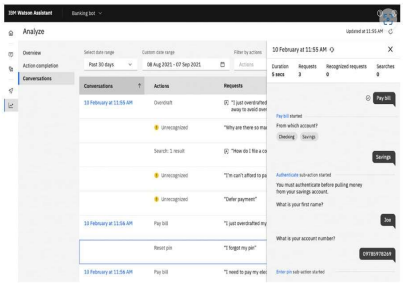
Dashboard (sample)



Goal Completion Rate (sample)



Blockers (Sample)



Conversation Logs (sample)

INTEGRATION & IMPLEMENTATION OVERVIEW

1. Exit criteria
   a. Application tested
   b. Residual defects acceptable for implementation
   c. Signoff upon the completion of the presentation from our course instructor
2. Deployment Strategy
   - Development Environment
     1. Kaggle - Model development
     2. Watson Assistant - Chatbot development
     3. Brackets - HTML CSS JS development
   - Github (Software development and version control)
     Flask Framework - Project Folder structure
     1. Static folder - CSS & Images
     2. Templates folder - HTML with Watson Chatbot scripts incorporated
     3. uploads folder
     4. Procfile
     5. Python code
     6. Model (.h5 file)
     7. requirement.txt
   - Heroku (Web Hosting)
3. Presentation for the project (MVP) and approval
4. Live - via Heruko Addon (Application Portfolio Manager) to deploy to public site, granting access to public

FLASK

… developed by Armin Ronacher based on Python

Flask is a Microweb Framework written in Python, it is a small and lightweight Python web framework

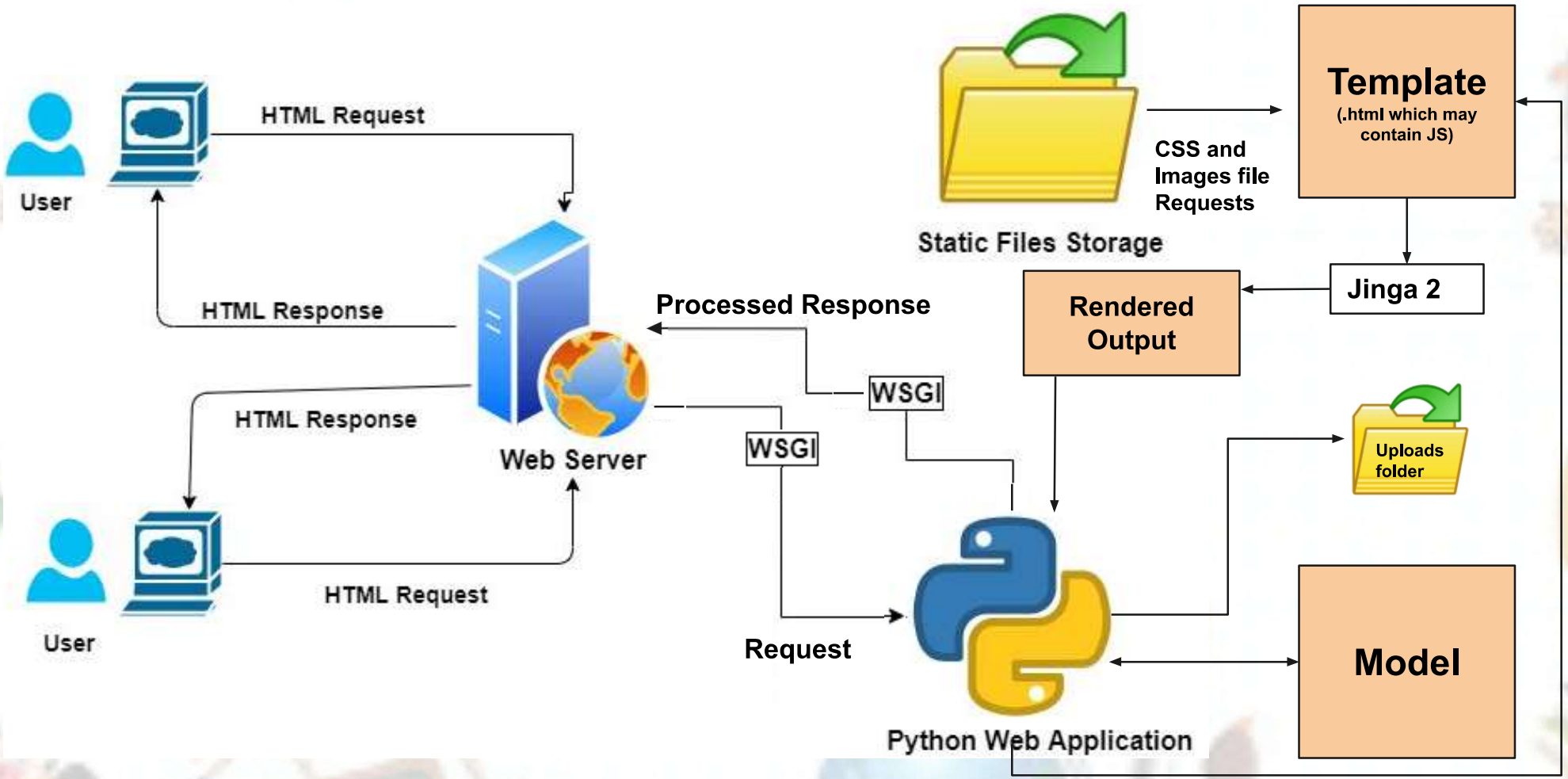
Advantages:

1. Not require a particular tool/library
2. Third party libraries can be added
3. Ease of setup and use
4. Freedom to build the structure of the web application

Components (Flask is a wrapper around these):

1. ItsDangerous - a safe data serialization library for Python
2. Jinja 2 - a template rendering engine
3. MarkupSafe - a string handling library for Python
4. Werkzeug - a utility library for Python for Web Server Gateway Interface (WSGI) which is a standard between web server & web apps

Application Request/Response Flow (End to End)

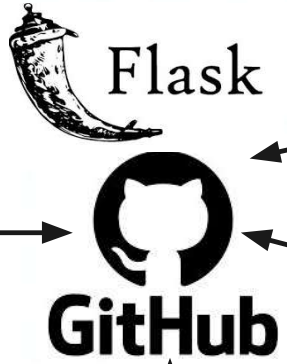


MVP

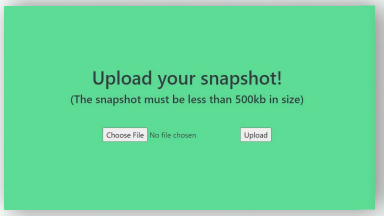
[redacted]

https://i-can-write.herokuapp.com/

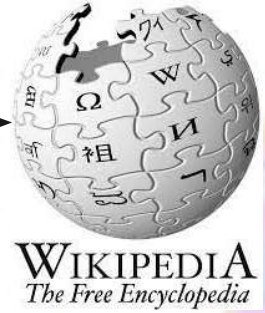
INTEGRATION & DEPLOYMENT



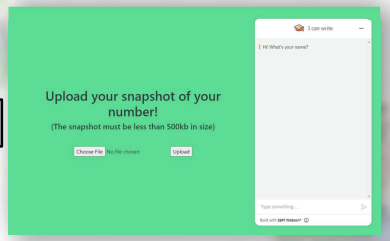
MVP 1.1

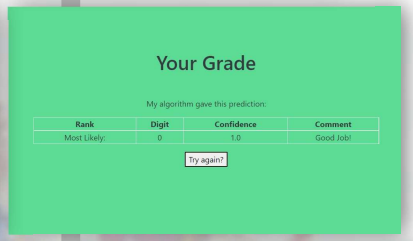


IBM Cloud Functions

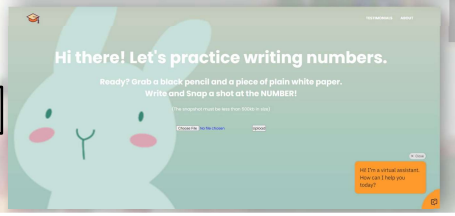


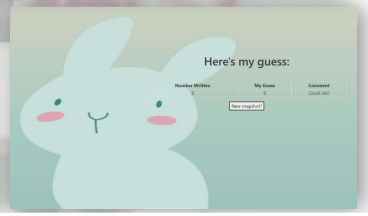
MVP 1.2

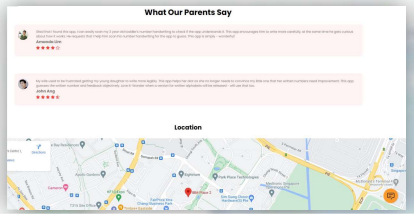


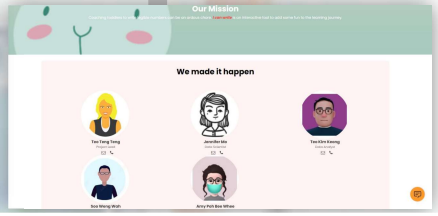


MVP 1.3

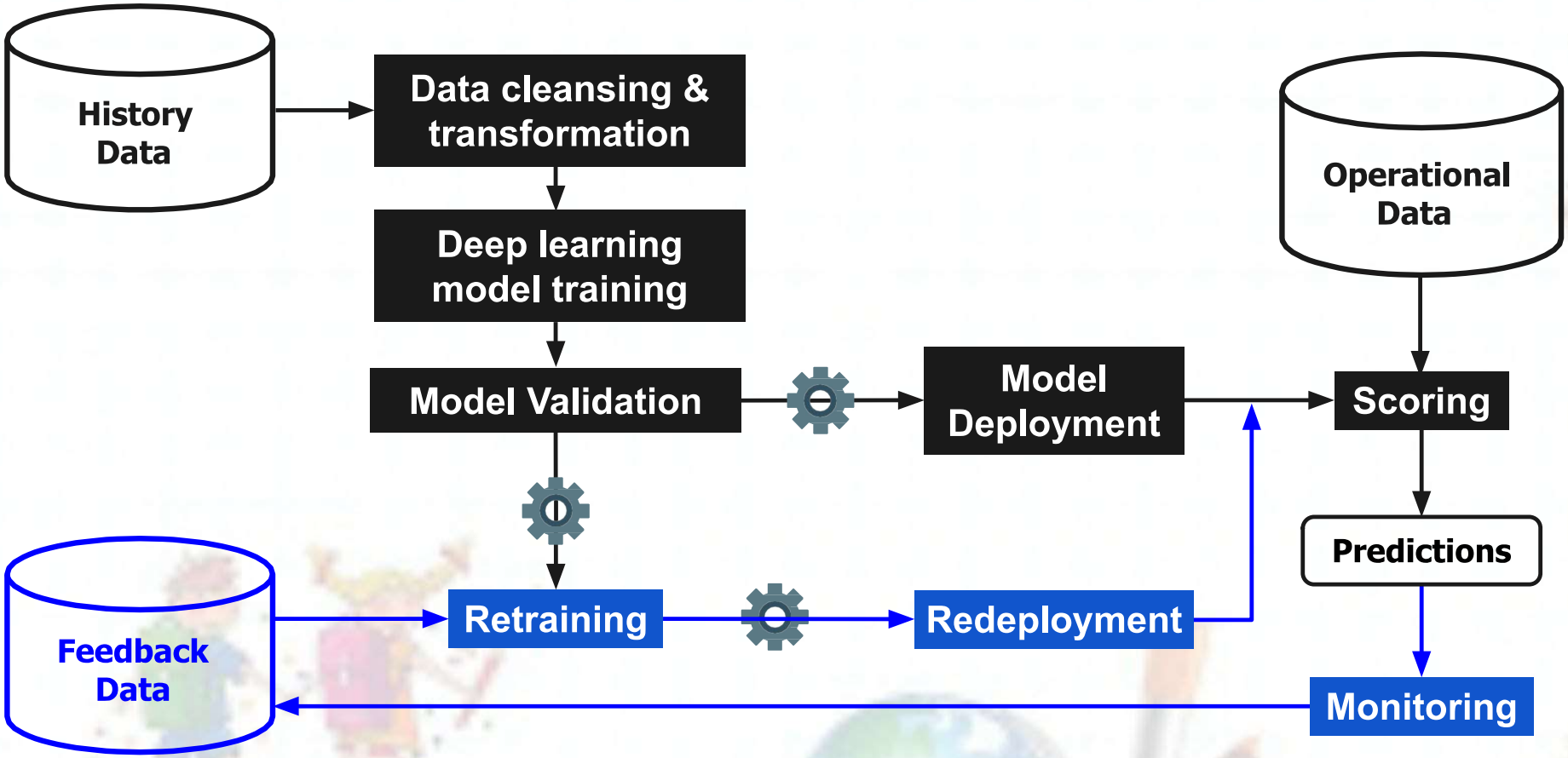








CONTINUOUS LEARNING



Project Learnings

CONSTRAINTS and CHALLENGES

Constraints

- Though fast & easy to develop & implement, the adoption of IBM cloud Watson Studio & AutoAI are not realised in view of limited CUH quotas.
- Github maximum upload-via- browser file size of 25MB limits model capability development.
- Heroku hard limit for compressed file size of 500MB limits web functionality development.
- App crashes when there are 5 HTML linkages. 4 HTML linkages are optimal.

Challenges

- Cumbersome manual integration updates among open-sourced services.
- Available digit datasets do not differentiate legible and illegible handwriting.
- Lack of expertise in html and python coding

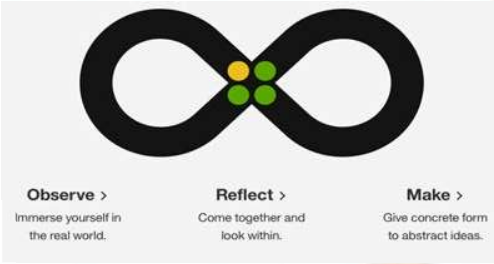
FUTURE ENHANCEMENTS

- Enhancement to model training & testing with colored (RGB) image dataset
- Enhancement to model training & testing with negative examples
- Expand problem statement to include alphabet and mother tongue character handwriting
- Chatbot and Model integration
- Chatbot and user feedback form enablement by setting up an email ID for TigerMDsAssociation.com

….. given larger repository file upload size limit

TAKEAWAYS

- As an extension to Enterprise design thinking, AI essential framework is applied to examine intent, understand data, reason application of AI to the problem and know its implications.



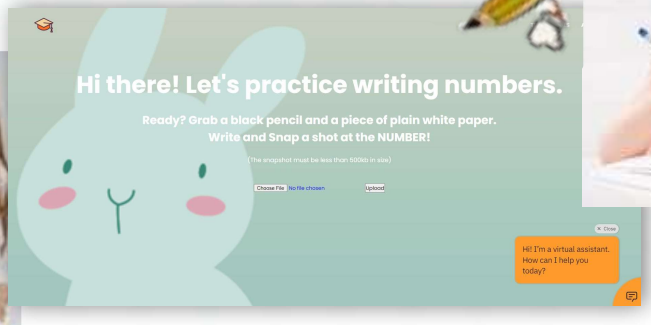
- Start small and fast deployment with MVP.

- Learnt integration within IBM Cloud environment, as well as IBM Cloud services with non-IBM (open-sourced) services.

- Nice-to-have differentiated legibility dataset.

TAKEAWAYS

I can write



- Demonstrated benefits of AI computer vision (image recognition) in relieving parent's pain points and development of child's pre-school number penmanship.

Thank You !

REFERENCE

- https://www.kaggle.com/code/melwin23/mini-project/notebook
- https://thinkinfi.com/upload-and-display-image-in-flask-python/
- https://pyimagesearch.com/2018/12/31/keras-conv2d-and-convolutional-layers/
- https://medium.com/analytics-vidhya/inverting-an-image-using-numpys-broadcasting-method-1f5beb7f9fa5
- https://www.tensorflow.org/api_docs/python/tf/keras/optimizers/Adam
- https://datascience.stackexchange.com/questions/44124/when-to-use-dense-conv1-2d-dropout-flatten-and-all-the-other-layers
- https://towardsdatascience.com/image-classification-in-10-minutes-with-mnist-dataset-54c35b77a38d
- Writing Numbers playlist by Inspired by Kinder https://www.youtube.com/playlist?list=PL8sp-x8dV_7Dy2tBFIV_1G76JGILocrPE
- https://www.udemy.com/courses/search/?courseLabel=6056&lang=en&q=Handwriting+for+kids&sort=relevance&src=ukw
- https://ably.com/topic/webhooks#using-webhooks-three-simple-steps
- https://www.esri.com/arcgis-blog/products/arcgis-enterprise/administration/webhooks-dev-summit-2019/
- 25+ Top Chatbot Statistics For 2022: Usage, Demographics, Trends (startupbonsai.com)
- IBM Watson Assistant - Chatbot Analytics | IBM

BACKUP SLIDES

DATASET: MNIST

- The MNIST (Mixed National Institute of Standards and Technology) database was introduced by LeCun et al. [2] in **1998**.
- Since then, this dataset has been widely used as a testbed for different machine learning and pattern recognition proposals.
- The MNIST database contains a total of 70,000 instances, from which 60,000 are for training and the remainder are for testing.
- The database comprises two different sources: NIST's Special Database 1 (collected among high-school students) and NIST's Special Database 3 (retrieved from Census Bureau employees).
- The training and test set were chosen so that a same writer would not be involved in both sets.
- The training set contains samples from more than 250 writers.

DATASET: EMNIST

- The source for building EMNIST database was NIST Special Database 19 (NIST SD 19), containing NIST's (National Institute of Standards and Technology of the US) entire corpus of training materials for handprinted document and character recognition, including over 800,000 manually checked and labelled characters from almost 3700 writers who filled a form.
- There are six different splits provided in this dataset. A short summary of the dataset is provided below:
  - EMNIST ByClass: 814,255 characters. 62 unbalanced classes.
  - EMNIST ByMerge: 814,255 characters. 47 unbalanced classes.
  - EMNIST Balanced: 131,600 characters. 47 balanced classes.
  - EMNIST Letters: 145,600 characters. 26 balanced classes.
  - **EMNIST Digits**: 280,000 characters. 10 balanced classes.
  - EMNIST MNIST: 70,000 characters. 10 balanced classes.

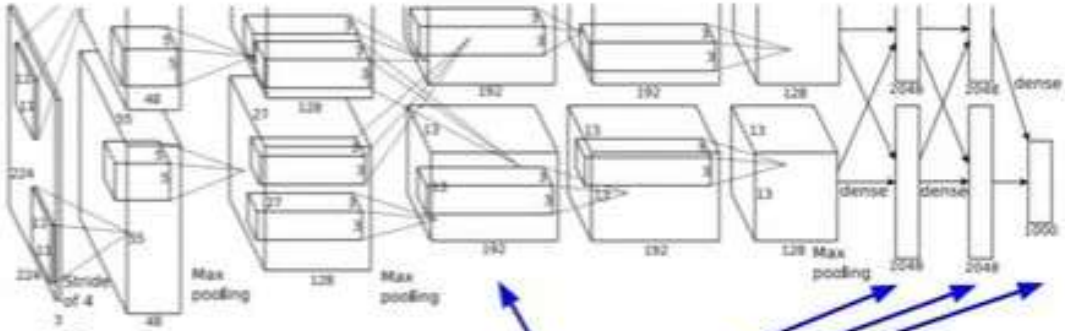
DATASET: Street View House Numbers (SVHN)

- SVHN is a real-world image dataset for developing machine learning and object recognition algorithms with minimal requirement on data preprocessing and formatting.
- It can be seen as similar in flavor to MNIST (e.g., the images are of small cropped digits), but incorporates an order of magnitude more labeled data (over 600,000 digit images) and comes from a significantly harder, unsolved, real world problem (recognizing digits and numbers in natural scene images).
- SVHN is obtained from house numbers in Google Street View images.

AI MODEL BUILDING: Our Reference

Case Study: AlexNet

[Krizhevsky et al. 2012]

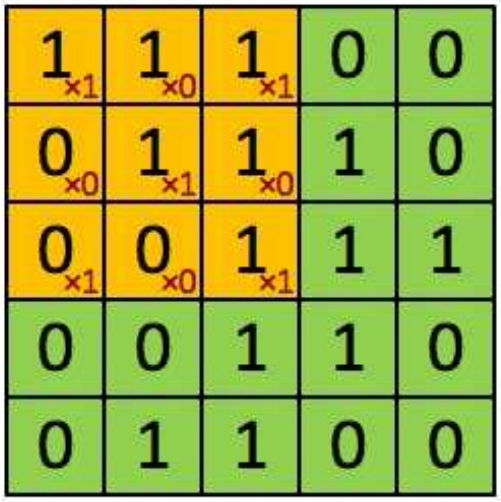


Full (simplified) AlexNet architecture:
[227x227x3] INPUT
[55x55x96] CONV1: 96 11x11 filters at stride 4, pad 0
[27x27x96] MAX POOL1: 3x3 filters at stride 2
[27x27x96] NORM1: Normalization layer
[27x27x256] CONV2: 256 5x5 filters at stride 1, pad 2
[13x13x256] MAX POOL2: 3x3 filters at stride 2
[13x13x256] NORM2: Normalization layer
[13x13x384] CONV3: 384 3x3 filters at stride 1, pad 1
[13x13x384] CONV4: 384 3x3 filters at stride 1, pad 1
[13x13x256] CONV5: 256 3x3 filters at stride 1, pad 1
[6x6x256] MAX POOL3: 3x3 filters at stride 2
[4096] FC6: 4096 neurons
[4096] FC7: 4096 neurons
[1000] FC8: 1000 neurons (class scores)

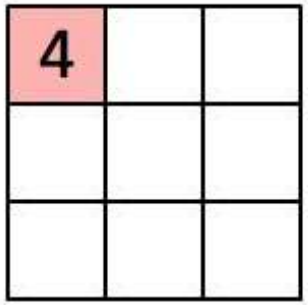
CONV3, FC6, FC7, FC8:
Connections with all feature maps in preceding layer, communication across GPUs

AI MODEL BUILDING: Feature Extraction

- https://miro.medium.com/max/526/0*Asw1tDuRs3wTjwi7.gif



Image



Convolved Feature

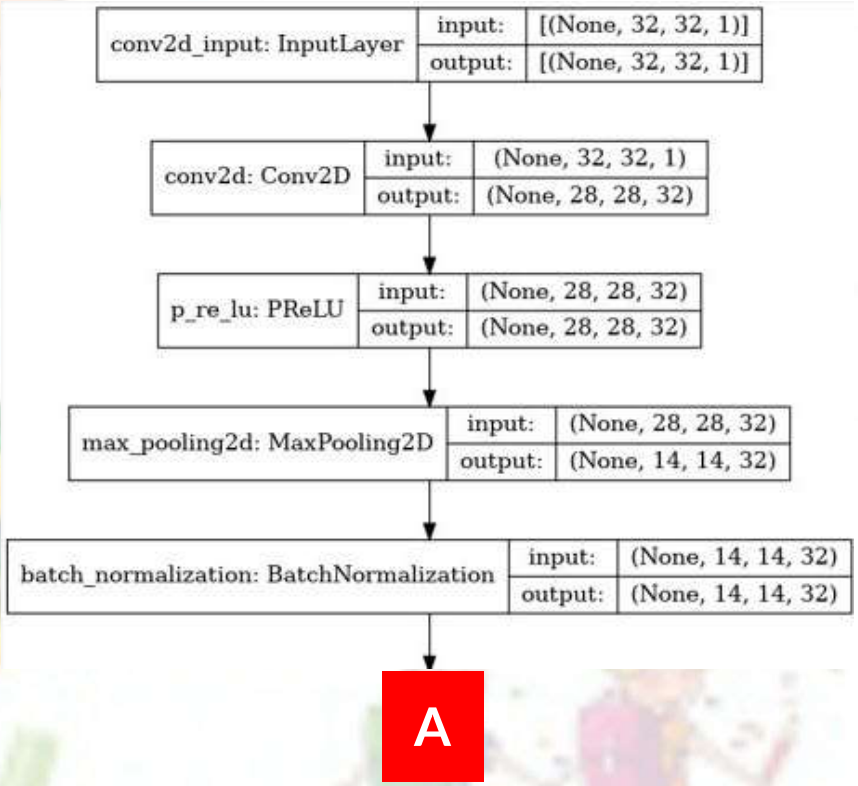
AI MODEL BUILDING: Model Summary

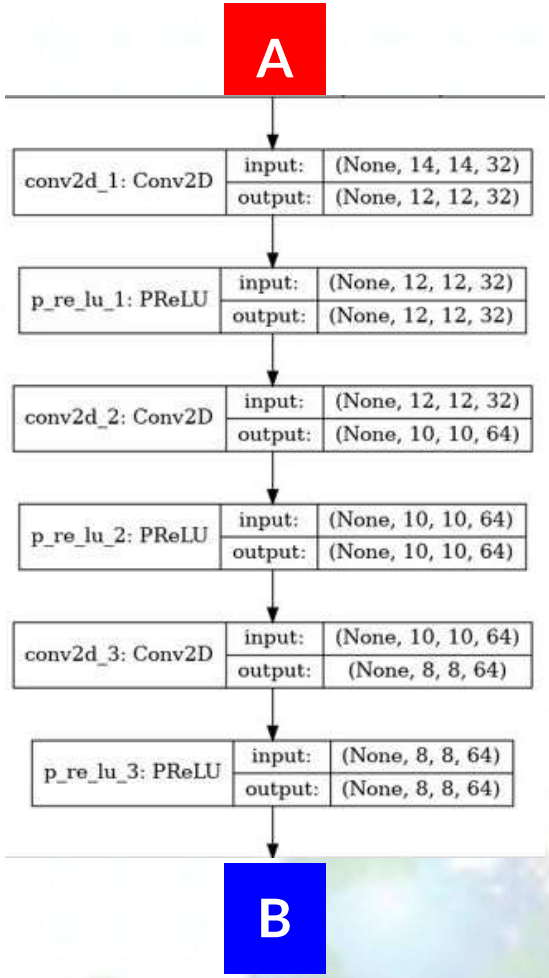
```
model = Sequential()

model.add(Conv2D(32,(5,5), kernel_initializer='he_uniform', input_shape=(32,32,1)))
model.add(tf.keras.layers.PReLU())
model.add(MaxPooling2D((2, 2)))
model.add(BatchNormalization())

model.add(Conv2D(32,(3,3), kernel_initializer='he_uniform'))
model.add(tf.keras.layers.PReLU())
model.add(Conv2D(64,(3,3), kernel_initializer='he_uniform'))
model.add(tf.keras.layers.PReLU())
model.add(Conv2D(64,(3,3), kernel_initializer='he_uniform'))
model.add(tf.keras.layers.PReLU())
model.add(MaxPooling2D((2, 2)))
model.add(Dropout(0.2))

model.add(Flatten())

model.add(Dense(128, kernel_initializer='he_uniform'))
model.add(tf.keras.layers.PReLU())

model.add(Dense(10, activation='softmax'))
```

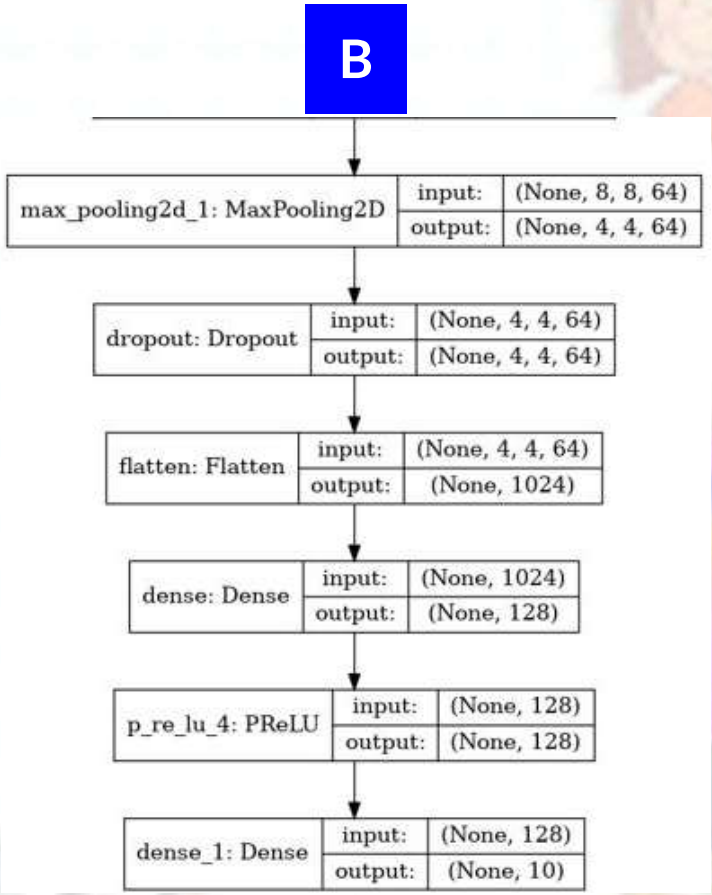
Layer (type)	Output Shape	Param #
conv2d (Conv2D)	(None, 28, 28, 32)	832
p_re_lu (PReLU)	(None, 28, 28, 32)	25088
max_pooling2d (MaxPooling2D)	(None, 14, 14, 32)	0
batch_normalization (BatchNo	(None, 14, 14, 32)	128
conv2d_1 (Conv2D)	(None, 12, 12, 32)	9248
p_re_lu_1 (PReLU)	(None, 12, 12, 32)	4608
conv2d_2 (Conv2D)	(None, 10, 10, 64)	18496
p_re_lu_2 (PReLU)	(None, 10, 10, 64)	6400
conv2d_3 (Conv2D)	(None, 8, 8, 64)	36928
p_re_lu_3 (PReLU)	(None, 8, 8, 64)	4096
max_pooling2d_1 (MaxPooling2	(None, 4, 4, 64)	0
dropout (Dropout)	(None, 4, 4, 64)	0
flatten (Flatten)	(None, 1024)	0
dense (Dense)	(None, 128)	131200
p_re_lu_4 (PReLU)	(None, 128)	128
dense_1 (Dense)	(None, 10)	1290

Total params: 238,442
Trainable params: 238,378
Non-trainable params: 64

AI MODEL BUILDING: plot_model







AI MODEL OPTIMIZATION: Activation Function