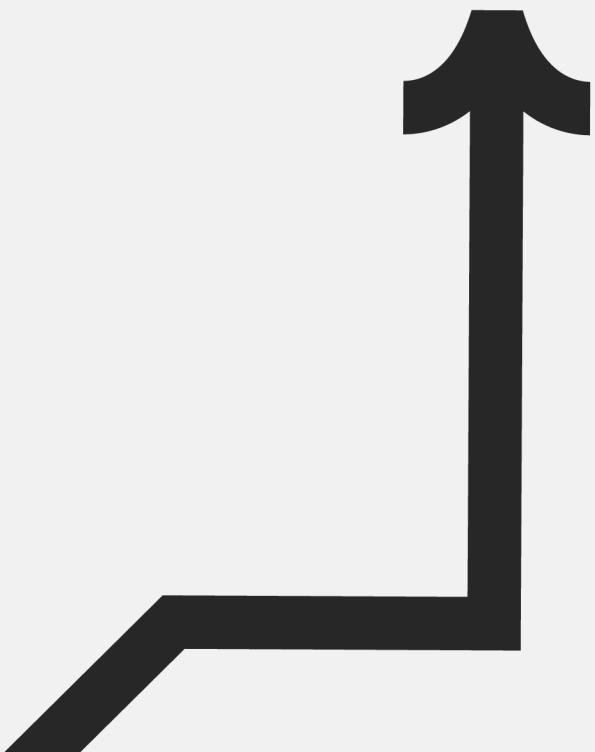
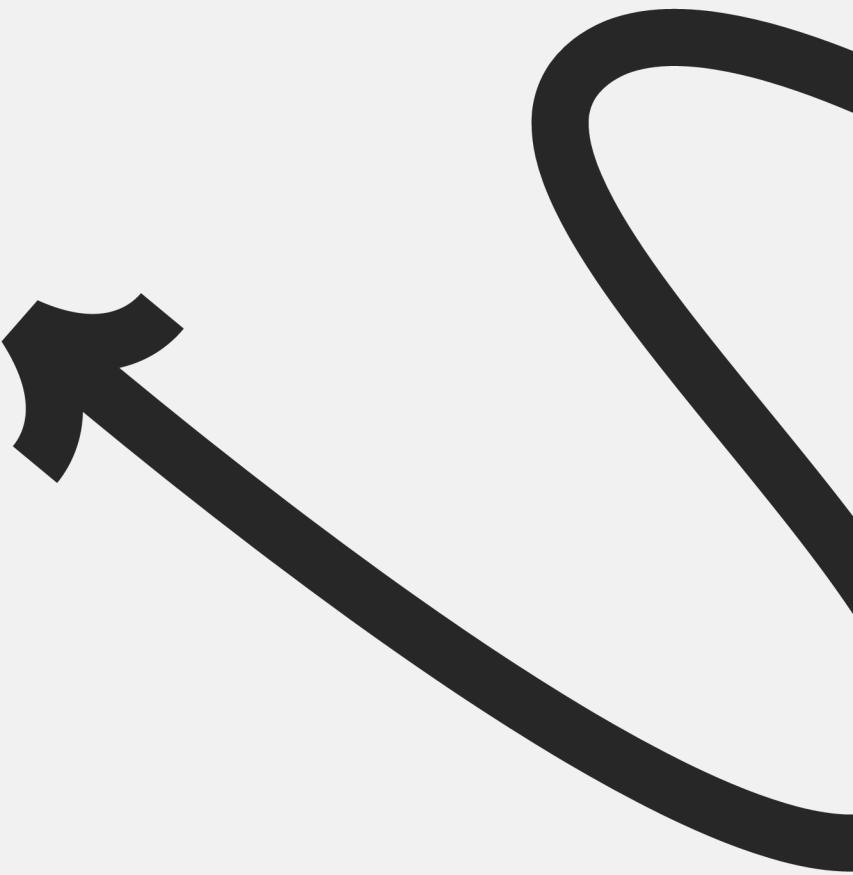


Project Presentation

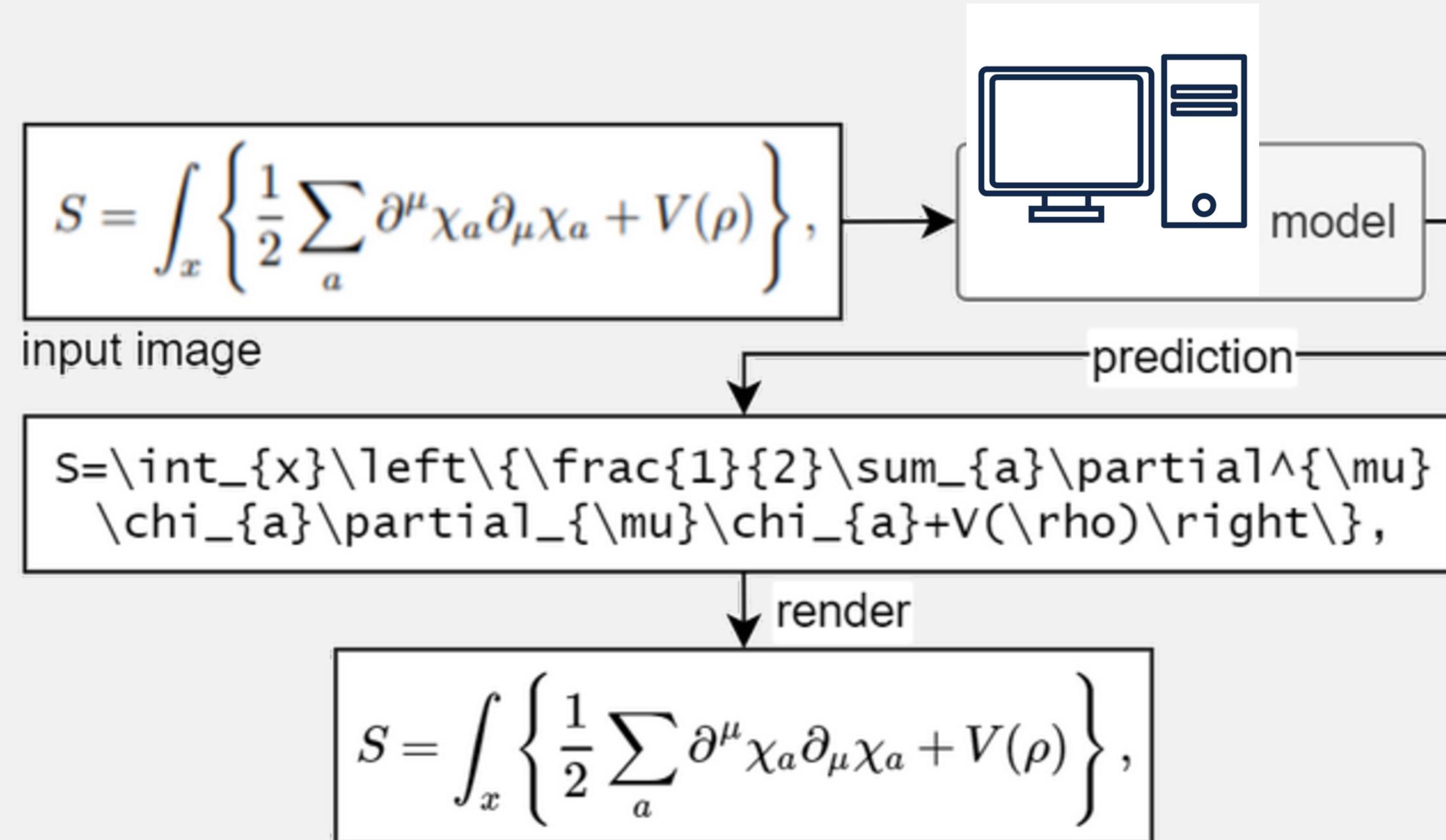
TEAM 19



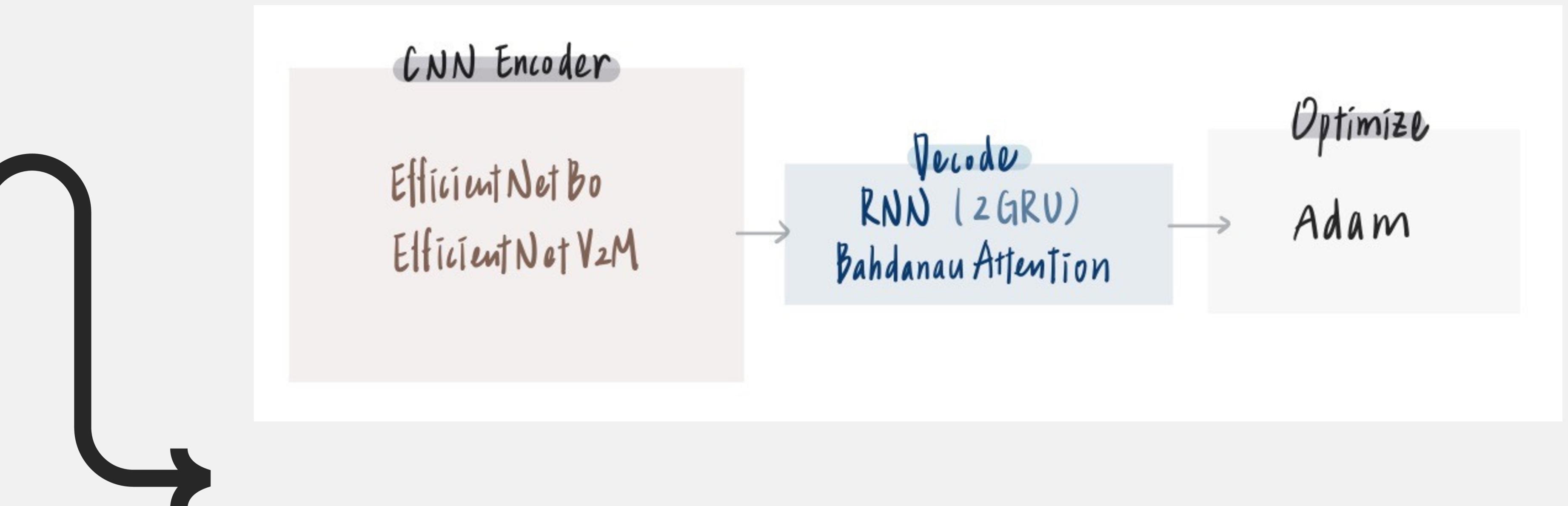
Intro



Project Intro



Model Flow



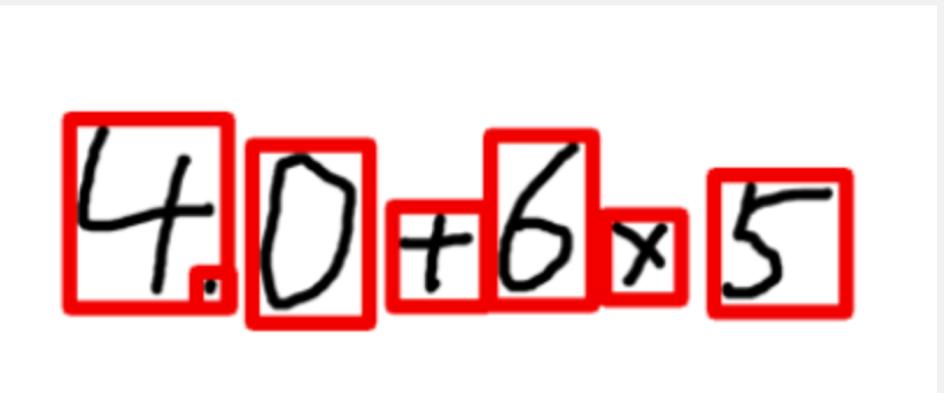
First version

Train a CNN model to recognize one single character in a small image.
As for bounding box? Simple CV technique would done.

Advantage : Easy to implement

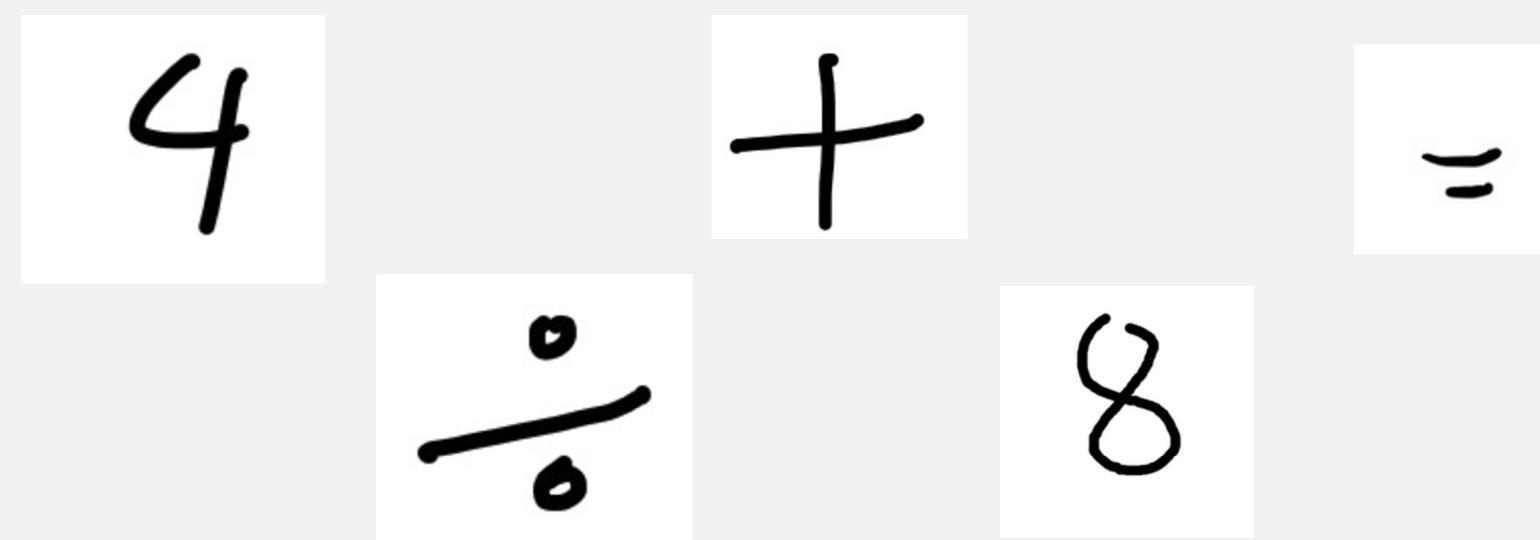
Disadvantage: Can't really tell the hidden logic , making it difficult to distinguish ‘(’ and ‘1”, “)” and “2” ...etc.

Limited model capability, hard to distinguish fraction, exponential.



Dataset

Handwritten math symbol and digit dataset (kaggle dataset)



Label: eight, equal ...etc

Improvement: RNN

Learn the dependency lies in the sequential data.

Not focusing on merely single character, try to summarize and comprehend the overall expression.

That is, turn the question into a image captioning problem.

Advantage: Better expressiveness, perform better over a meaningful sequence.

Disadvantage: Takes more data to train, higher complexity of the predicted expression leads to below expected expression

Dataset CROHME 2016

$$\cos 2\alpha$$

$$(y')^{(1)} = (x) \left(\frac{y}{\bar{y}}\right)$$

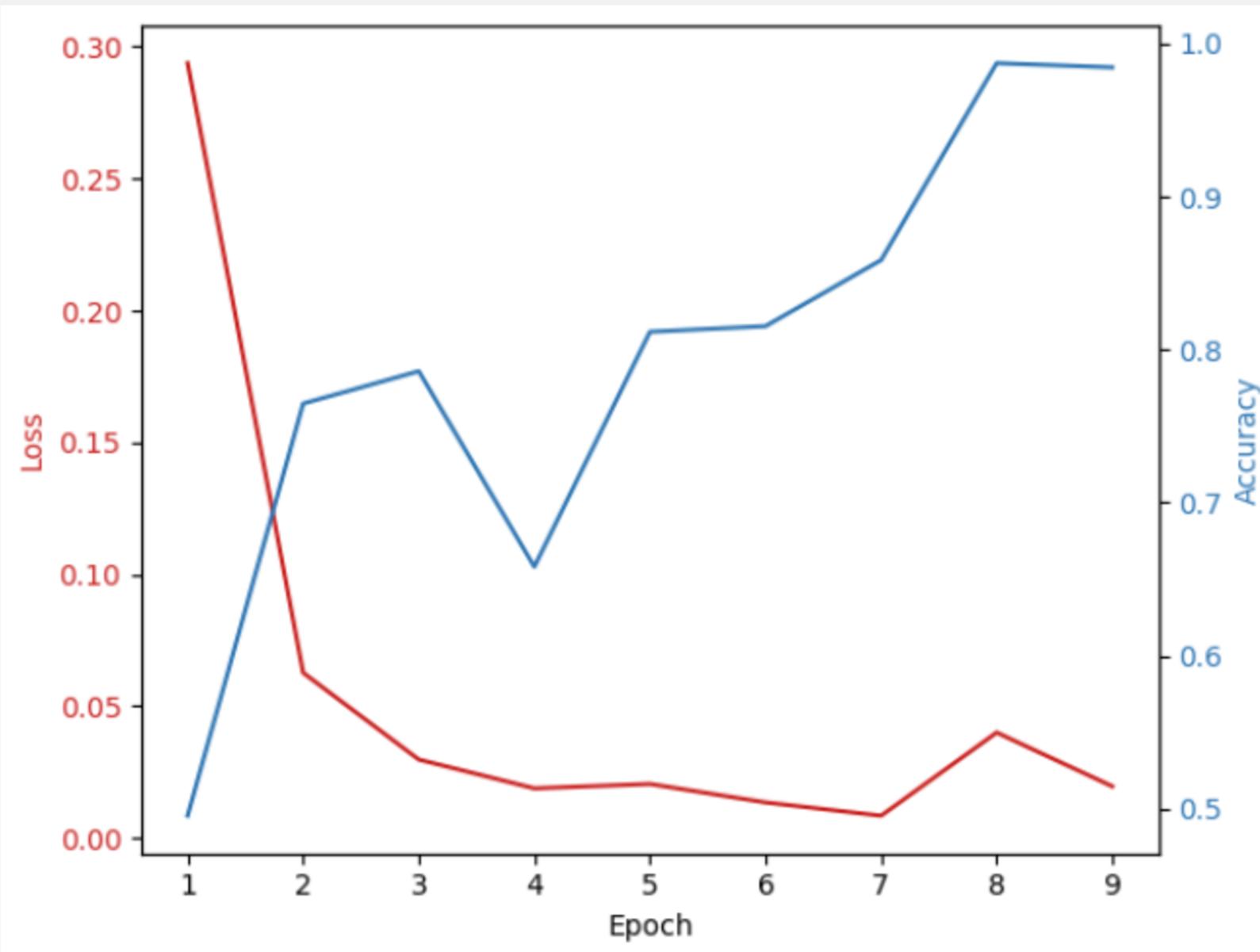
$$\frac{az^{-1}}{(1 - az^{-1})^2}$$

Label : $\sqrt{a} = 2^{-n} \sqrt{4^n a}$



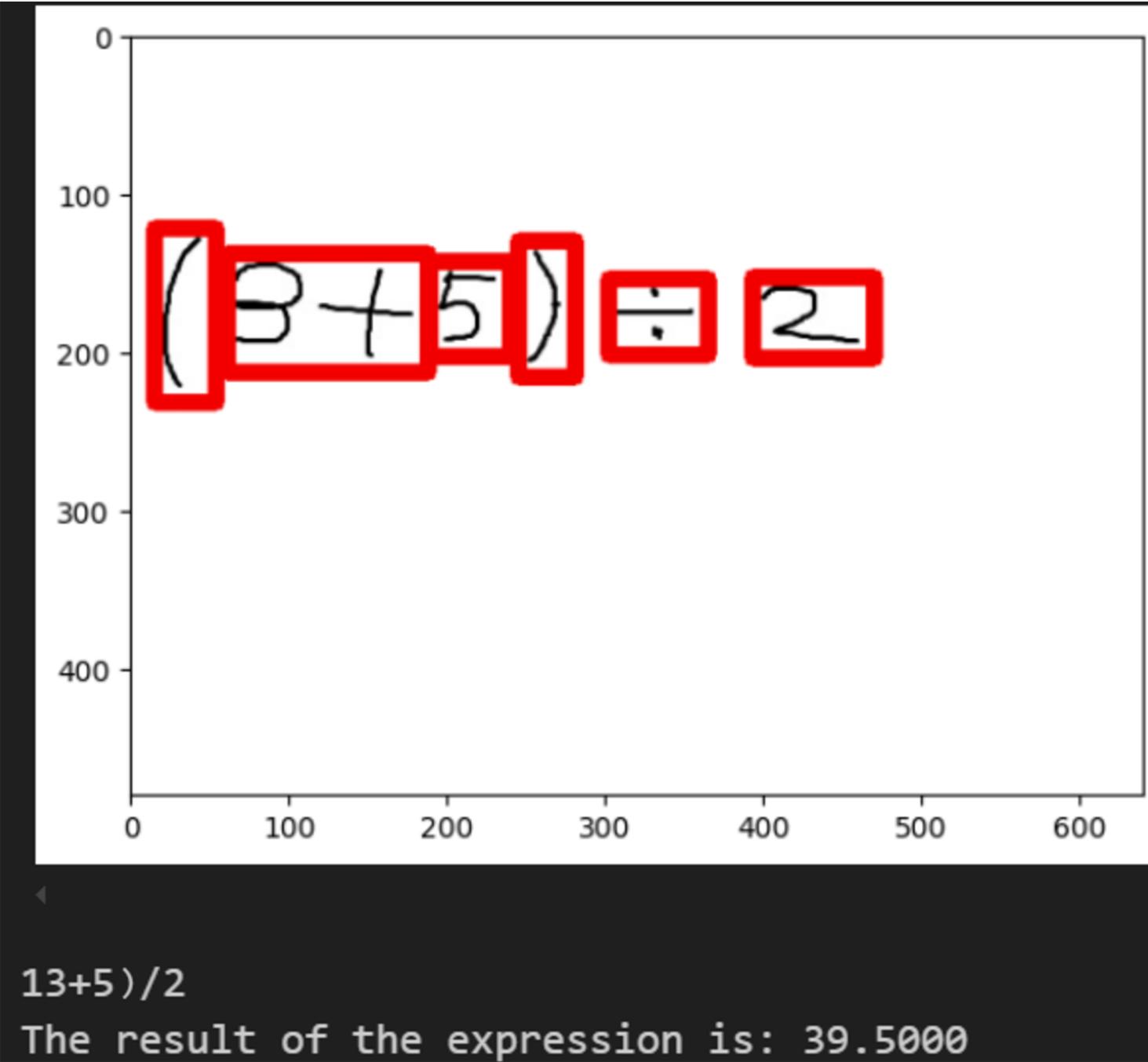
Simple CNN

Only evaluate over single character recognition



Accuracy: 0.984375
Only test on single character recognition
Not good at all in real complex expression

Performs fabulously over the testing set.
However, in real world application, it is quite useless.
Limited complexity, only accepted daily four operations of arithmetic.
For sequential data, it is more likely to mispredict similar characters based on the feature of single char, not observing the hidden logic.



Poor performance when facing daily expressions



RNN (2-Layer GRU)

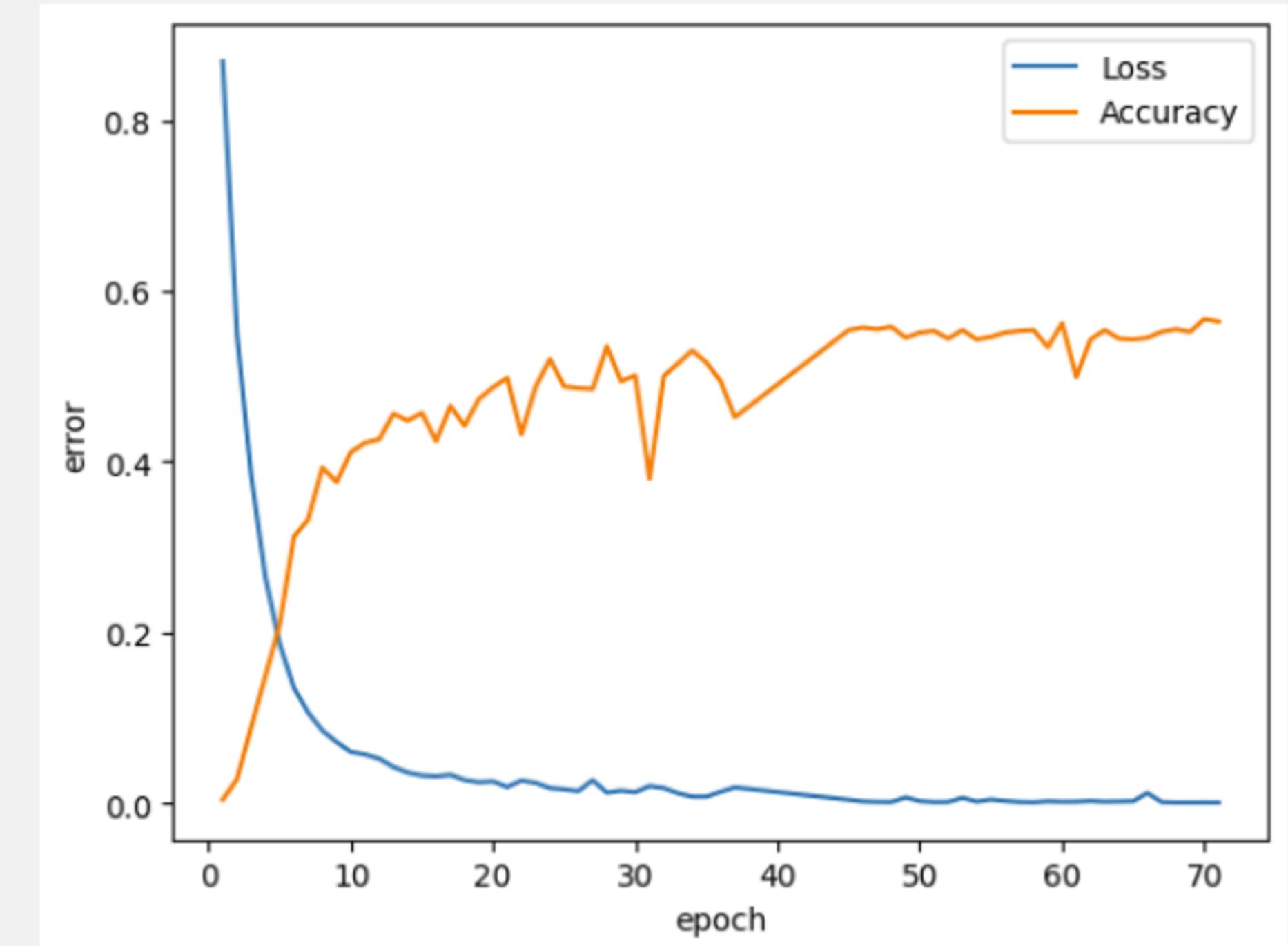
Error Curve

Accuracy: 0.575

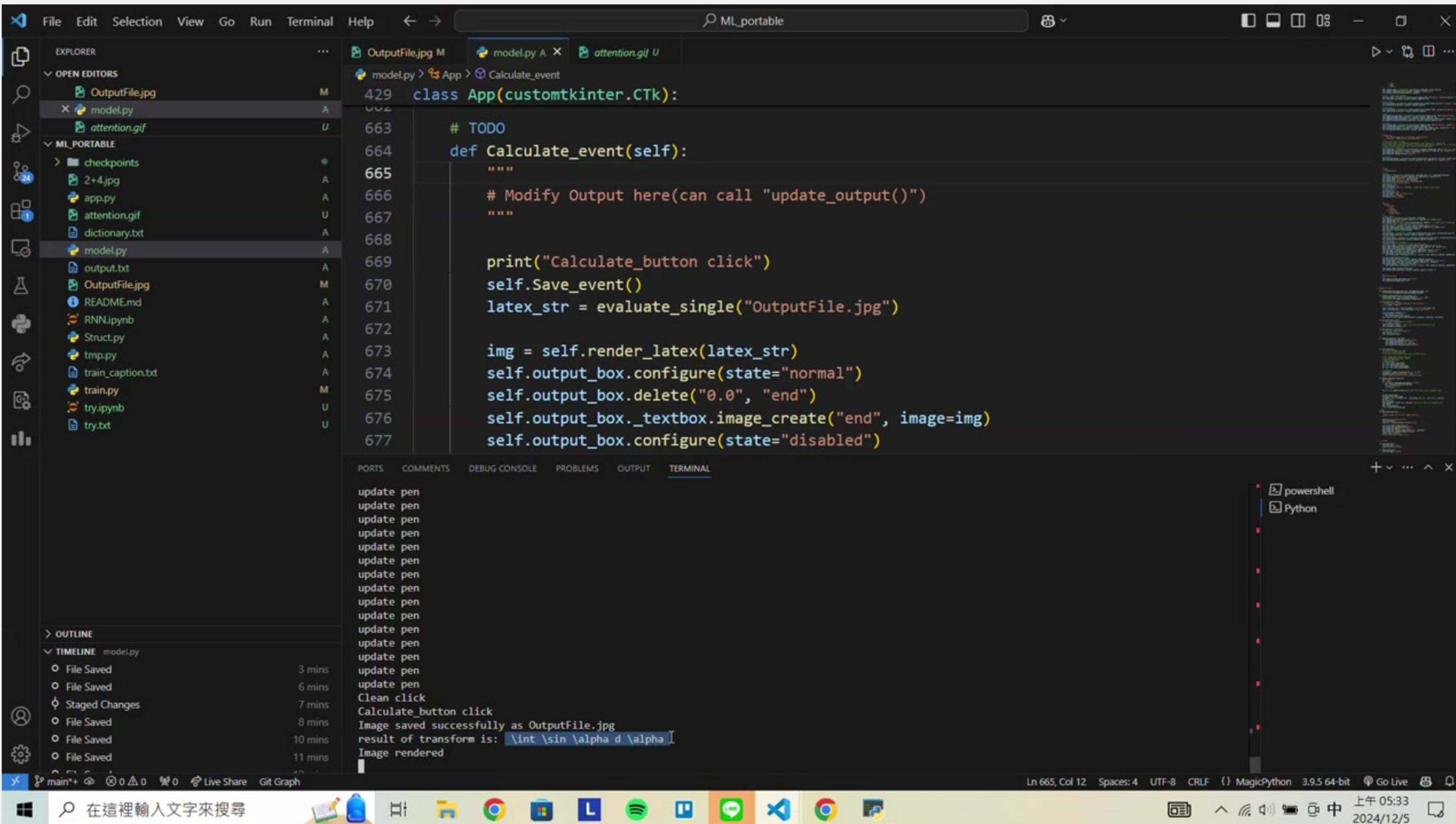
Define: Accuracy = 1 - CER

CER = Character Error Rate
(Convert to (0,1))

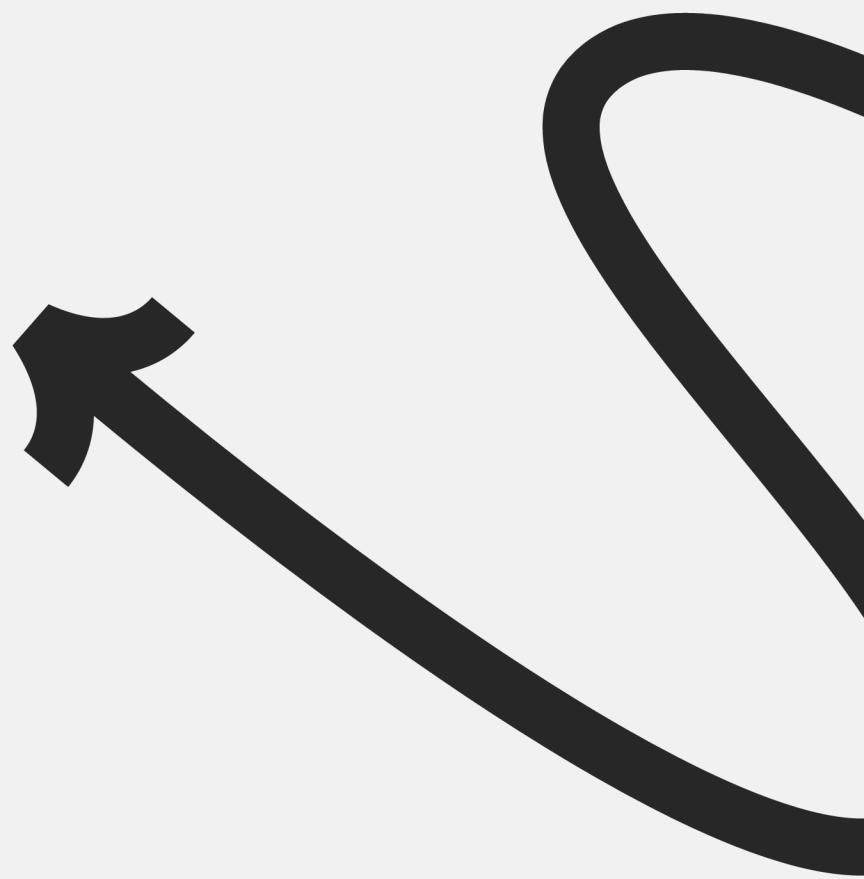
$$C.E.R = \frac{(Total.\ Char.\ Errors)}{(Total.\ Characters)} * 100$$



Demo Video - Expression (RNN)



QA



Question 1

It seems this project has great potential to become a lazy calculator, could possible connect this with a service such as Wolfram Mathematica and camera to calculate the answers for complex equations.

Solver: `latex2sympy2`

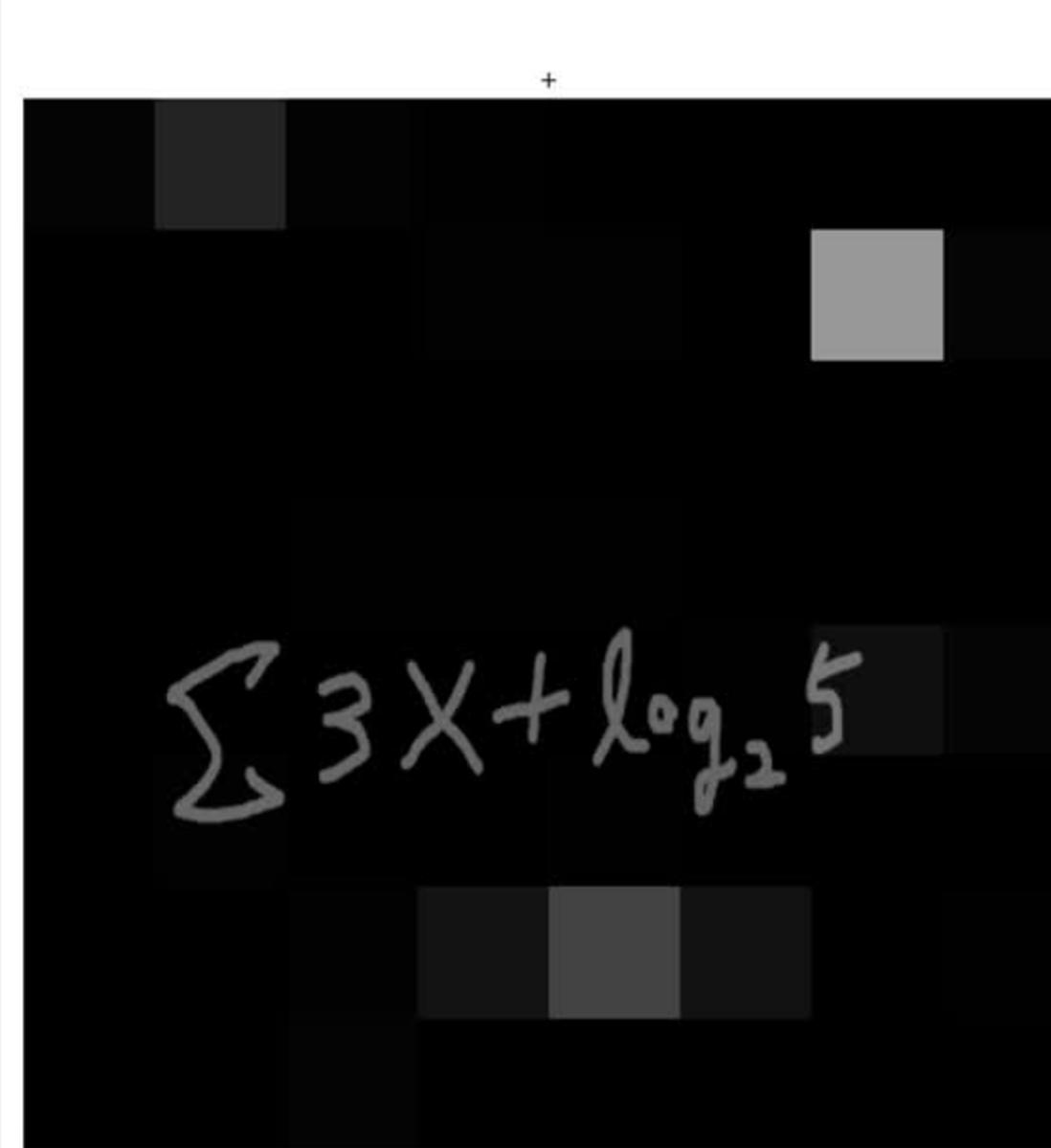
Plot: `numpy, matplotlib`



Question 2

Why the accuracy in the plot for RNN part is worse than CNN ?

Is the attention mechanism help improve your model?



Question 3

Does the Kaggle dataset include diverse handwriting styles?

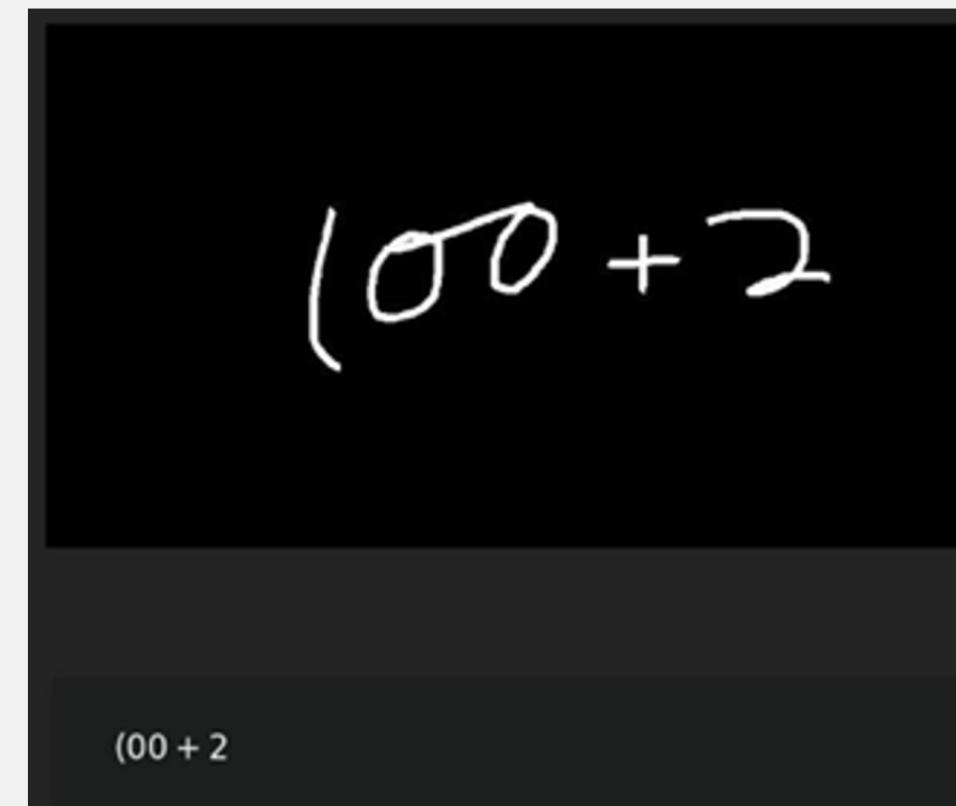
Since handwriting varies across different countries or individuals, this could affect the model's generalization ability.

For ambiguous characters ("1" and "l" or "-" and "_"), has contextual comparison been incorporated to infer the correct symbol based on surrounding context ?



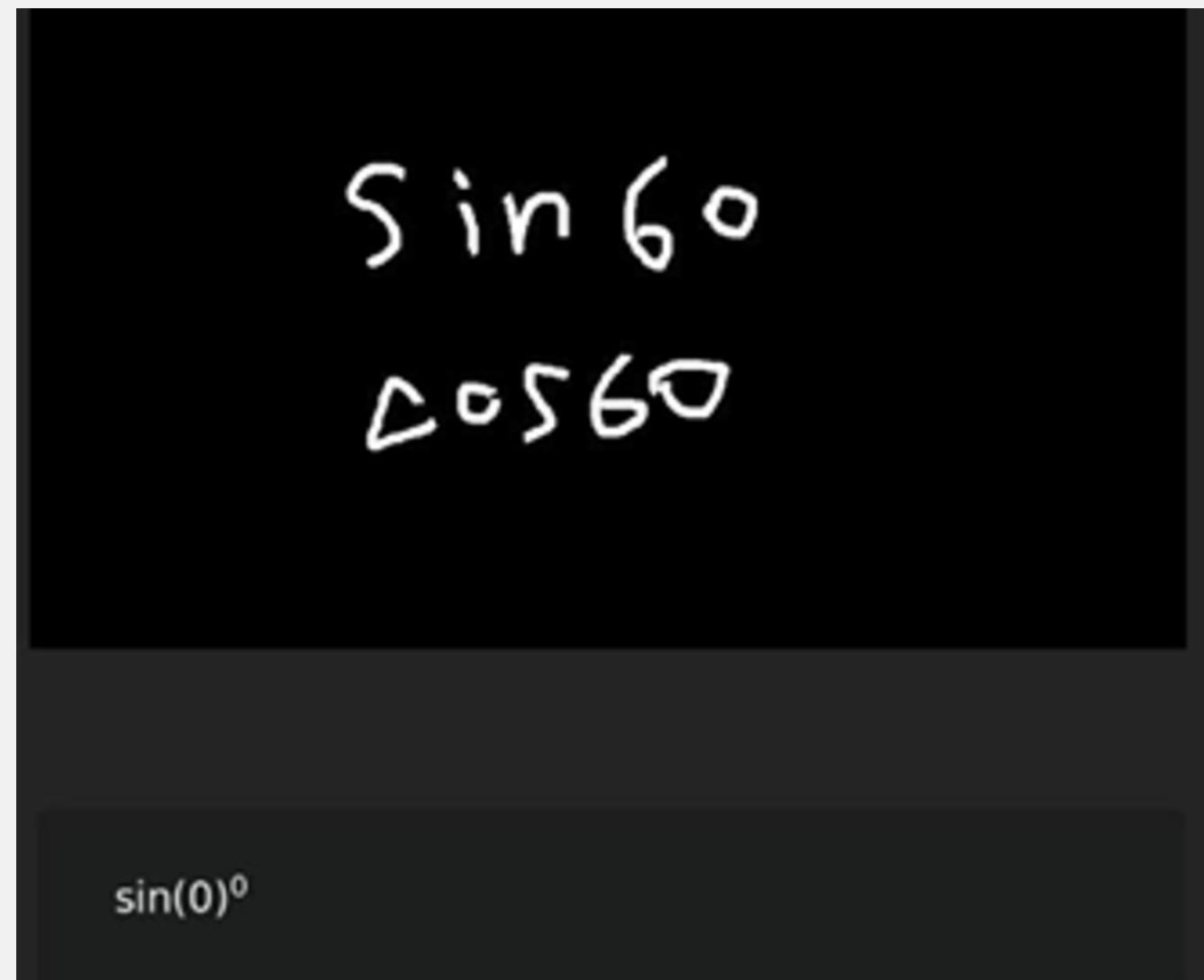
Question 4

If someone writes numbers very quickly—such as writing the number ""10"" with a connecting line between the ""1"" and the ""0""—will the model still correctly recognize it as ""10""? Or does the model not account for this scenario?



Question 5

What's the result of your model if you write multiple lines of equations?



End

TEAM 19

