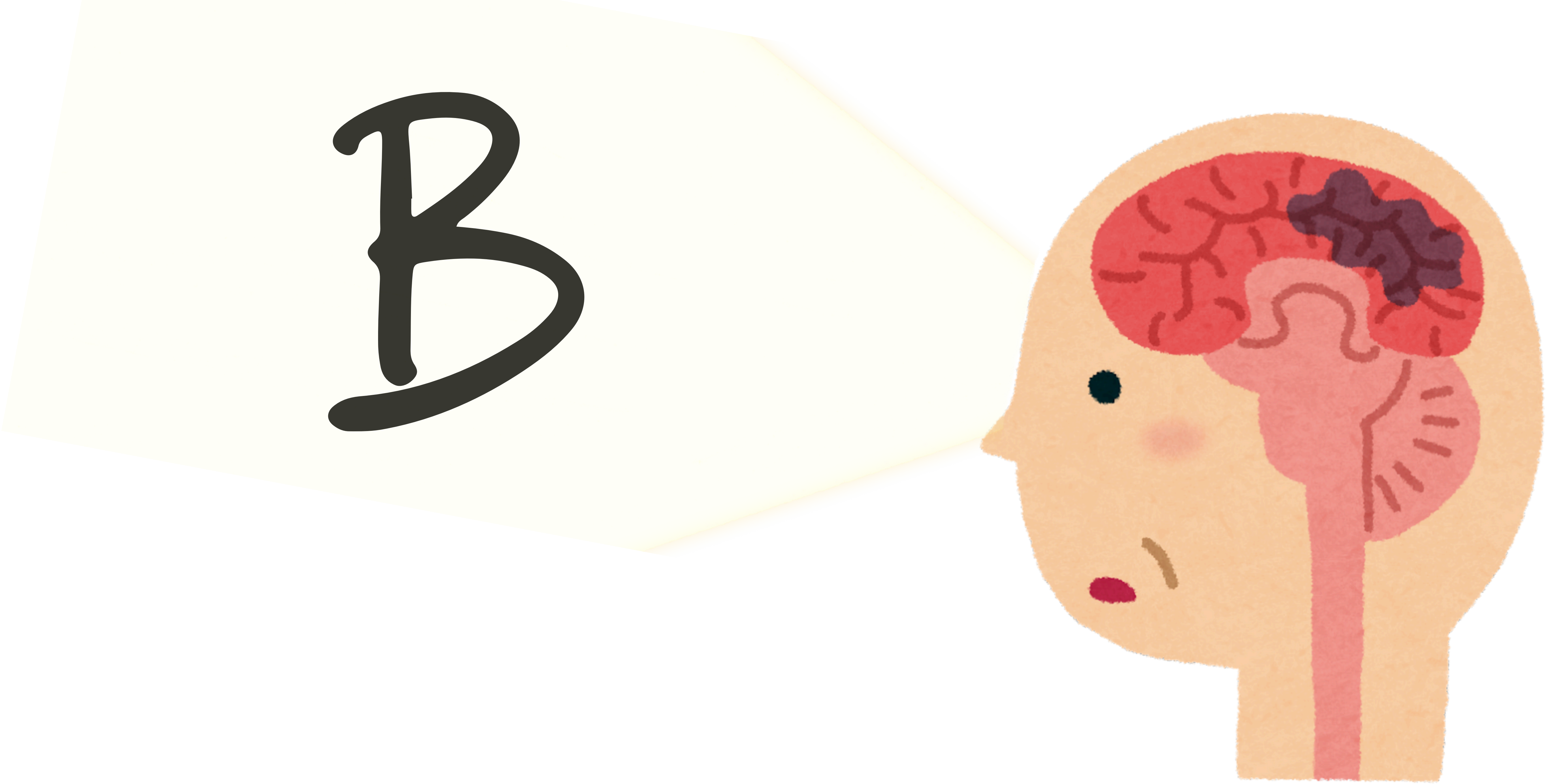
Handwritting recognition

using Convolutional Neural Networks (CNNs)

Presented by: Giang Nguyen & Ky Anh Le

01 Introduction

Can you recognize what is this character?



Can the computer recognize what is this character?

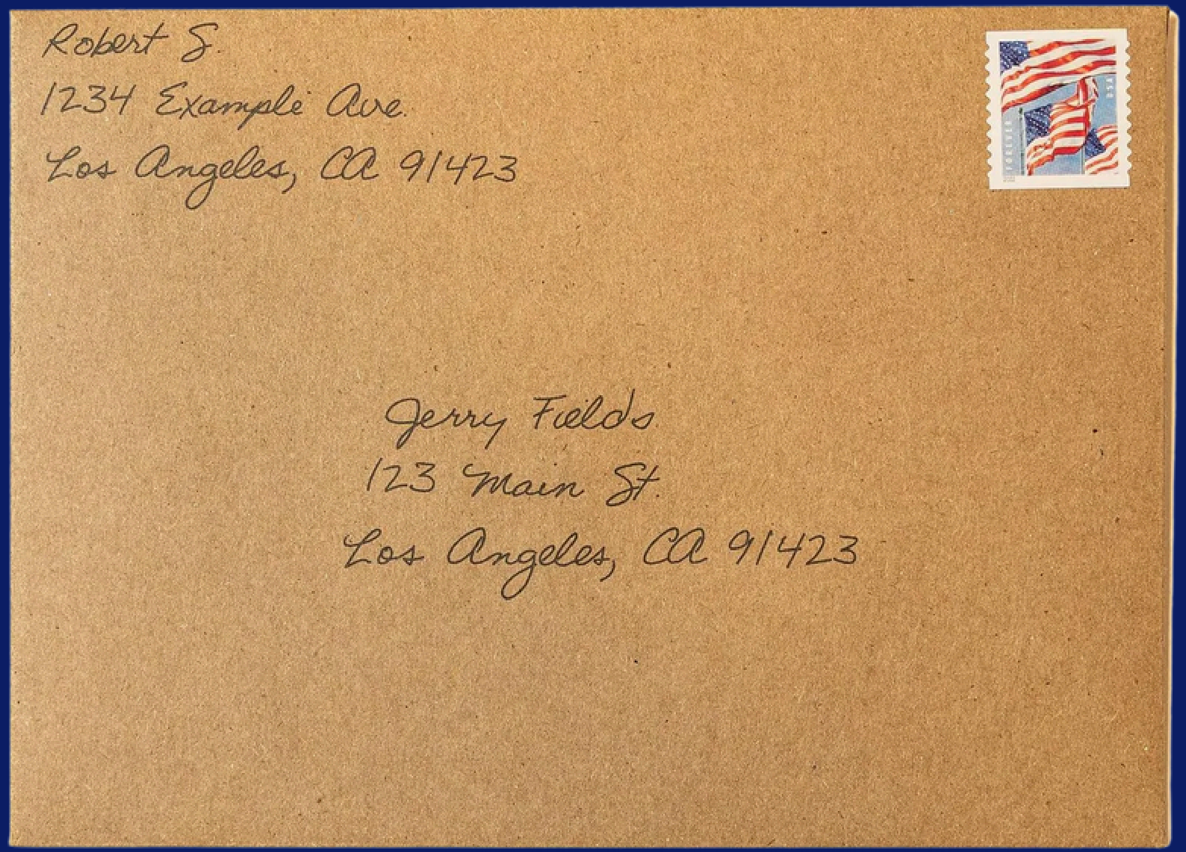
Handwritten Character Recognition is one of the core problems of computer vision and artificial intelligence, because it:

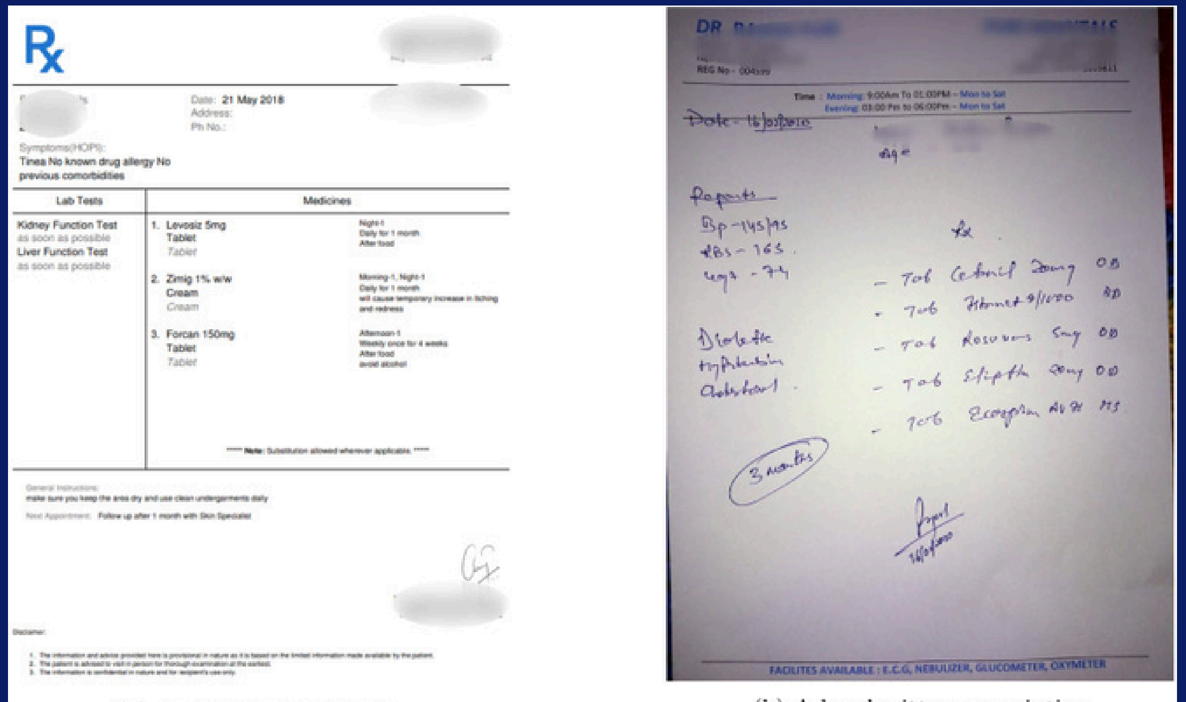
- Connects directly between regular human vision and digital systems.
- Is the first step in **automating data entry, document processing, and text understanding.**

However, **there are some key challenges:**

- Each person have **unique handwritting**
- **Image noise**: smudged ink, light, blurry scans.
- Identifying **different alphabets**

Practical Applications





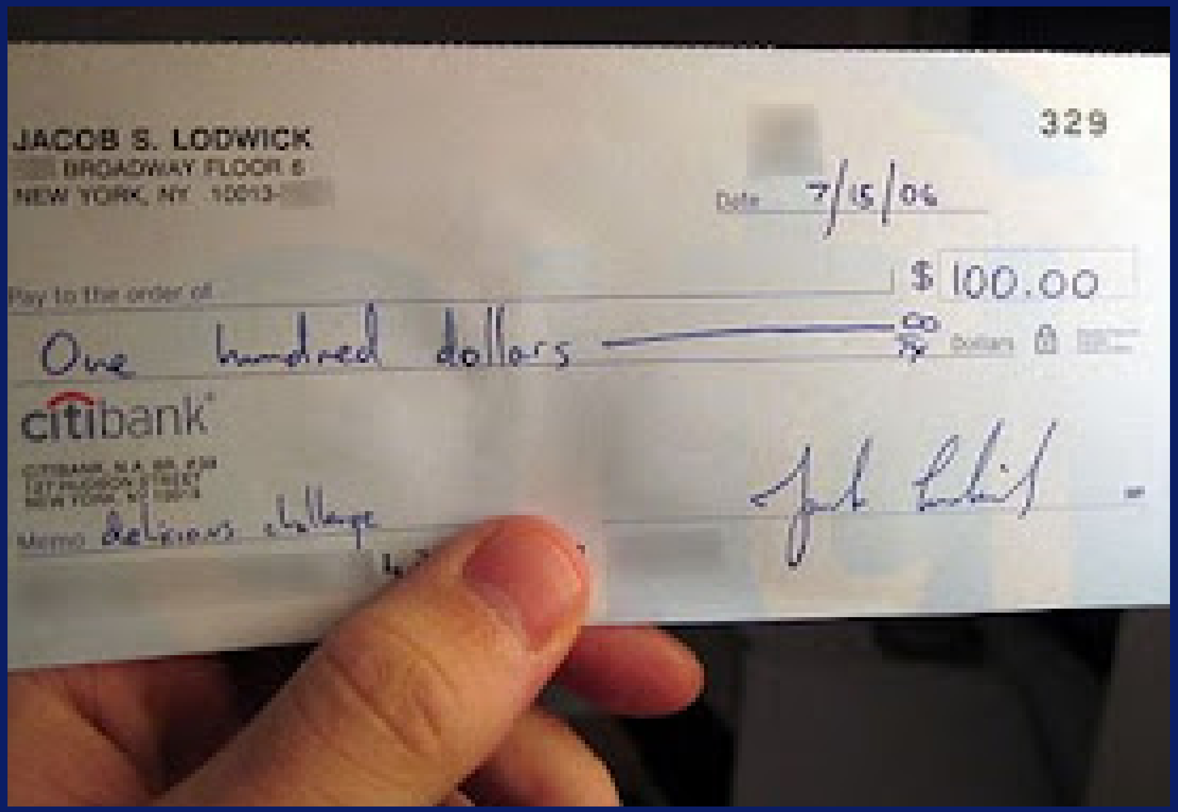
(a) A printed prescription (b) A handwritten prescription

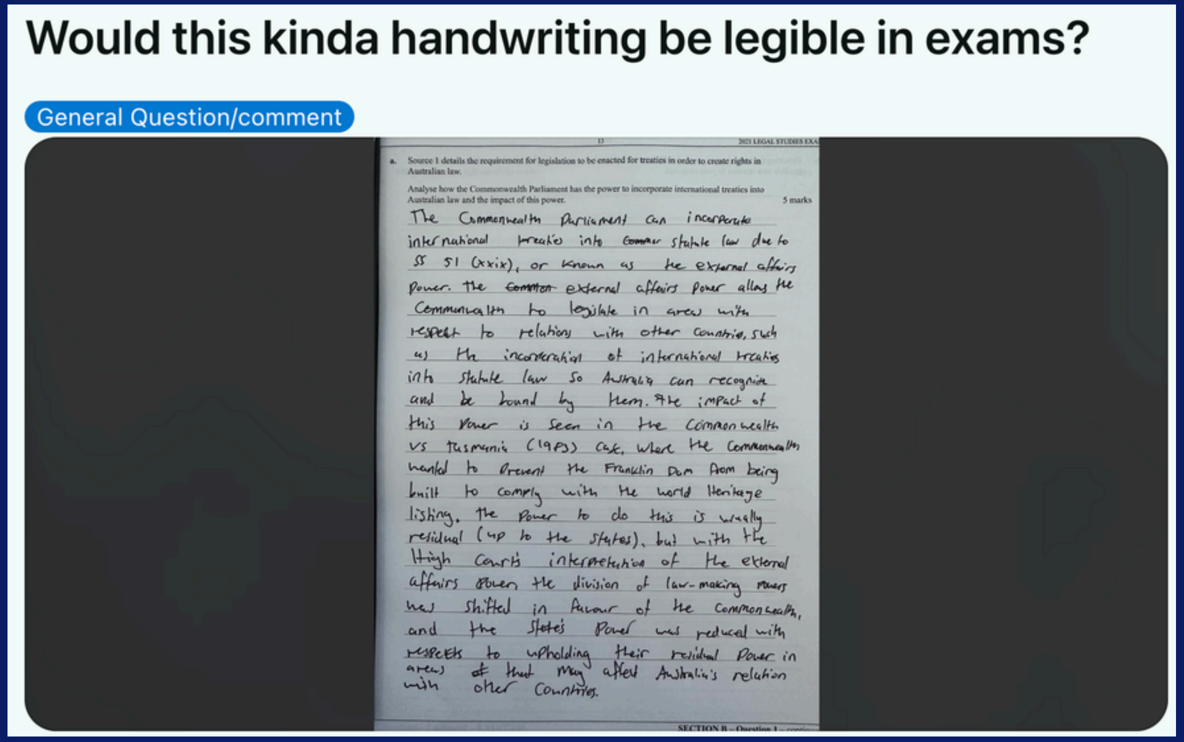
Postal Services: Recognizing handwritten addresses on envelopes.

Banking: Automated reading of handwritten checks.

Healthcare: Digitization of handwritten prescriptions.

Education: Automated grading of handwritten exams or homework.

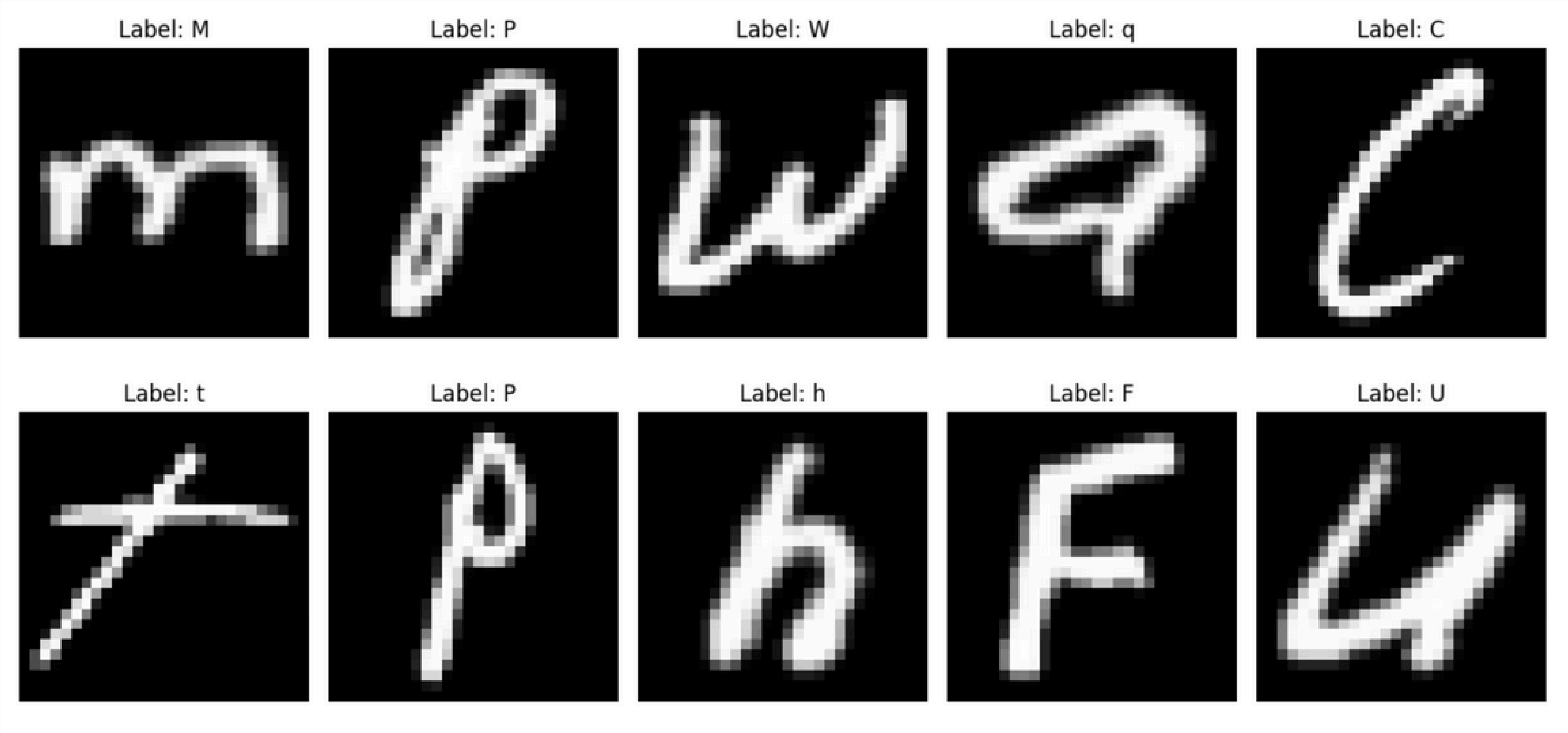




Used dataset:

The EMNIST dataset is a **set of handwritten character and digits derived** from the **NIST Special Database** 19 and converted to a 28x28 pixel image format and dataset structure that directly matches the MNIST dataset. Fufrther information on the dataset contents and conversion process can be found in the paper available at https://arxiv.org/abs/1702.05373v1.

Balanced - EMNIST dataset contains both character and digit **(47 classes)**



02 Deep learning approach

Convolutional Neural Network (CNN) is a specialized variant of Artificial Neural Network (ANN), **specifically designed to process spatially structured data, such as images.**

CNNs learn to **extract important features in images t**hrough **convolutional layers**, rather than requiring humans to manually design them.

Why CNN?

CNN learns the feature that Handwriting Image is Local - small image sample, with curves, edges, and horizontal and vertical lines, thanks to **convolution and pooling layers.**

No complex preprocessing required - In traditional methods, manual feature engineering is required. CNN learns features from image data, **simplifying the pipeline.**

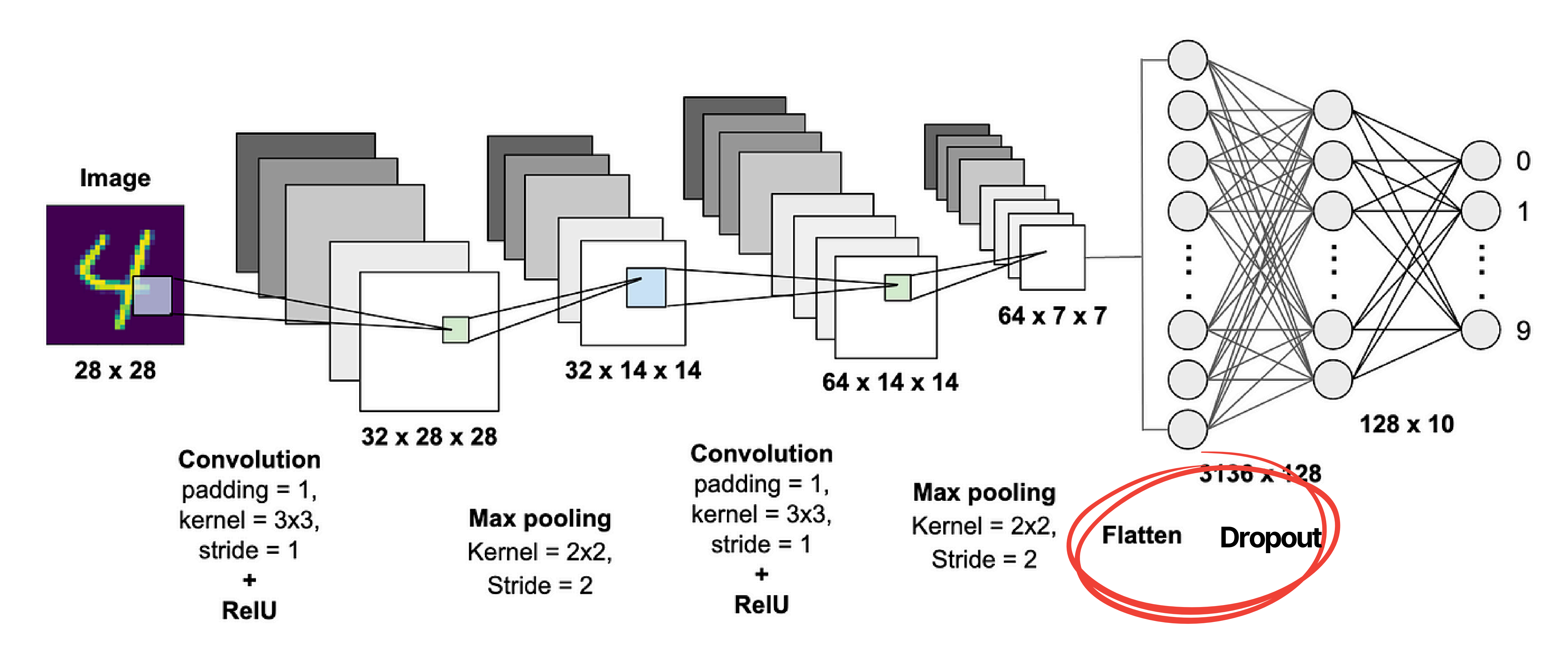
Invariance to small displacements and distortions - Handwritten letters are often misaligned, tilted, and resized.

CNNs learn features that are stable to small changes thanks to **pooling and weighted filters.**

03 Model Architecture

Main Layers:

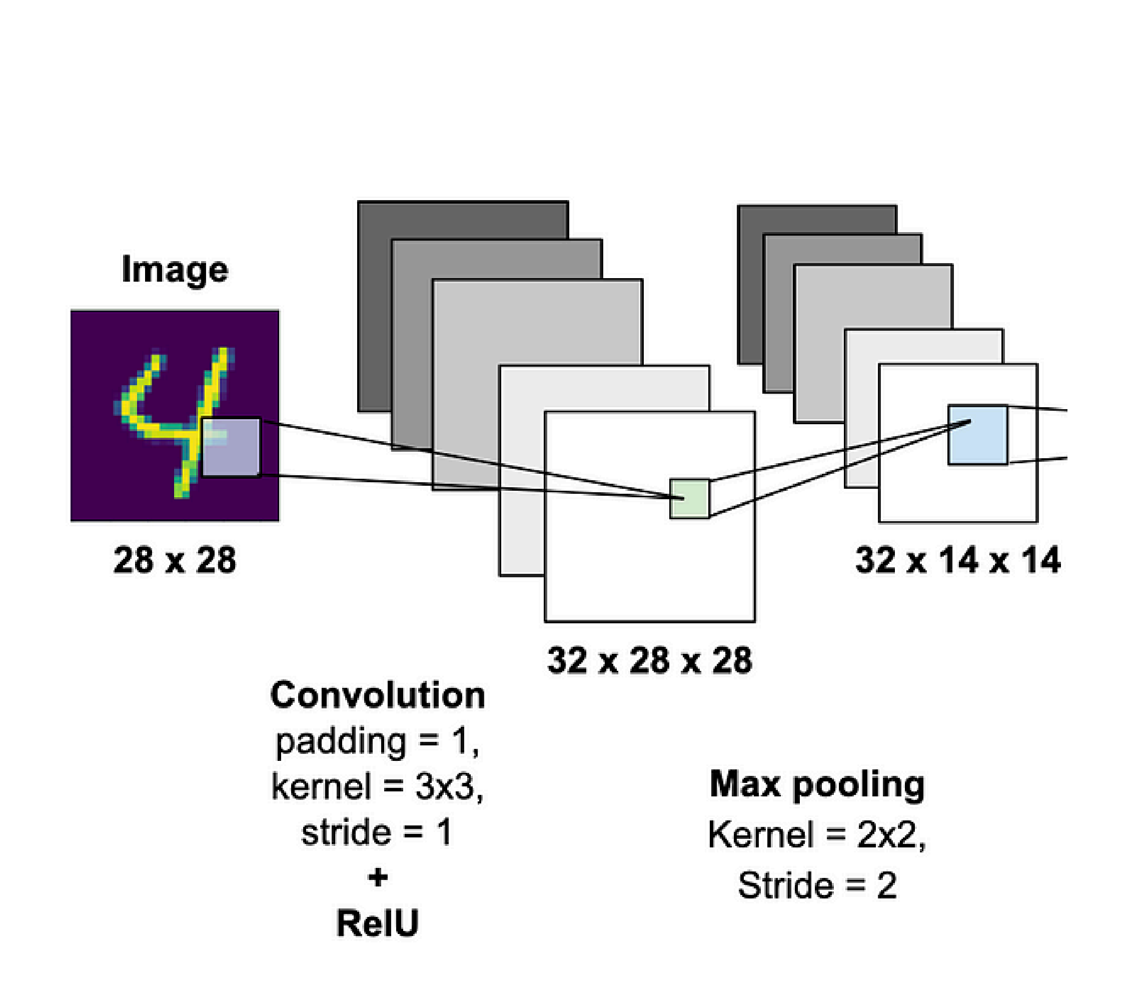
- **Conv2D (2 layers):** Extracting spatial features (edges, corners) from handwritten images
- **MaxPooling2D (2 layers):** Reduce spatial dimensionality to speed up and reduce overfitting
- **Flatten:** Convert 2D images into 1D vectors to feed into fully connected layers
- **Dense (2 layers):** Classification based on extracted features
- **Dropout:** Reduce overfitting by randomly "turning off" some nodes



Conv2D layer (32 filters, 3x3 kernel)
This is the first signature class – it is important to **“see” the basic structure of handwriting.**
Detect basic features like edges, **curves** from **original images** and Convo is **position invariant**

MaxPooling2D (2x2)
Highlight important features, ignore unnecessary details helps reducing feature size, make model compact, avoid overfitting.

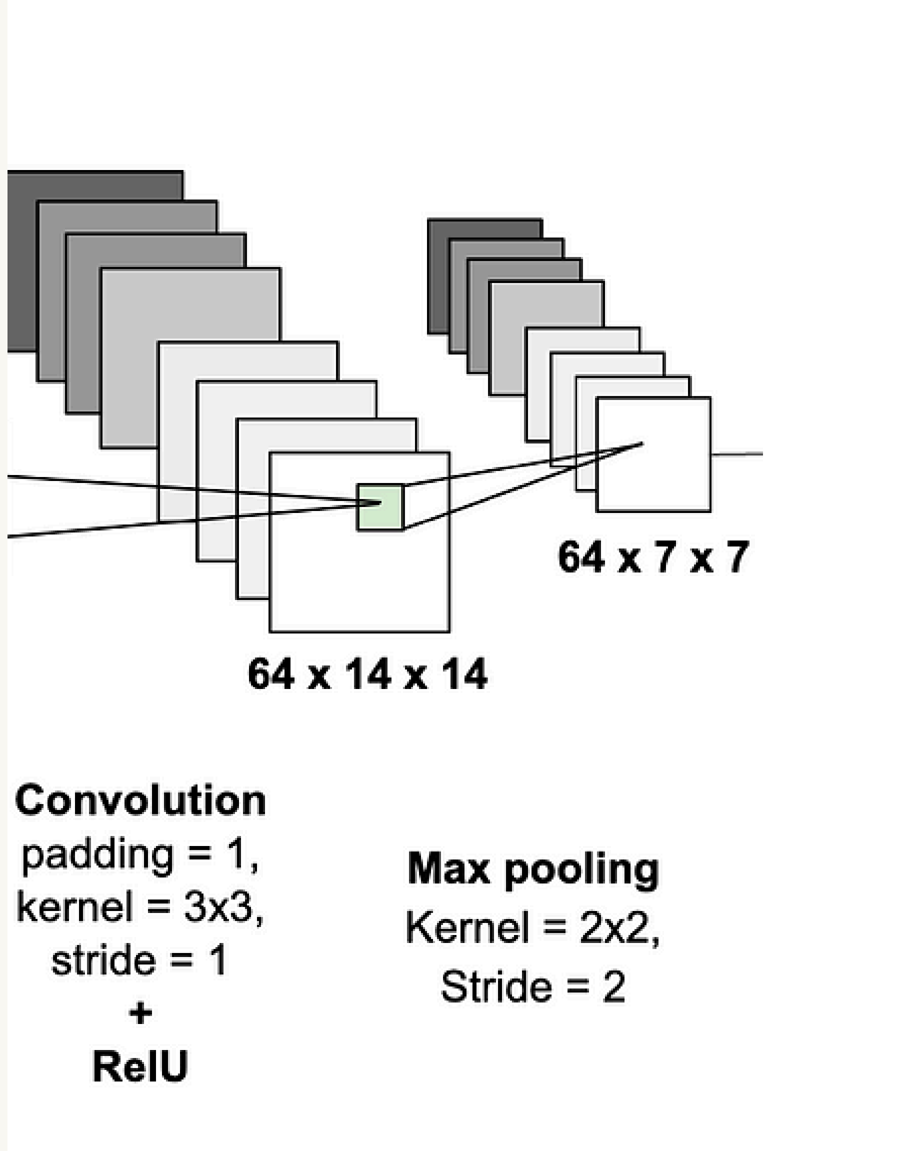
As we see from a distance, keep only the most important features of the character.



Conv2D (64 filters, 3x3 kernel)
Helps **extracting more complex features** like intersections, swirls.

Once the image is smaller, the model starts to recognize deeper shapes.

MaxPooling2D (2x2)
Continue to **reduce the dimensionality**, **retaining** the **most important information** to include in the **classification**.

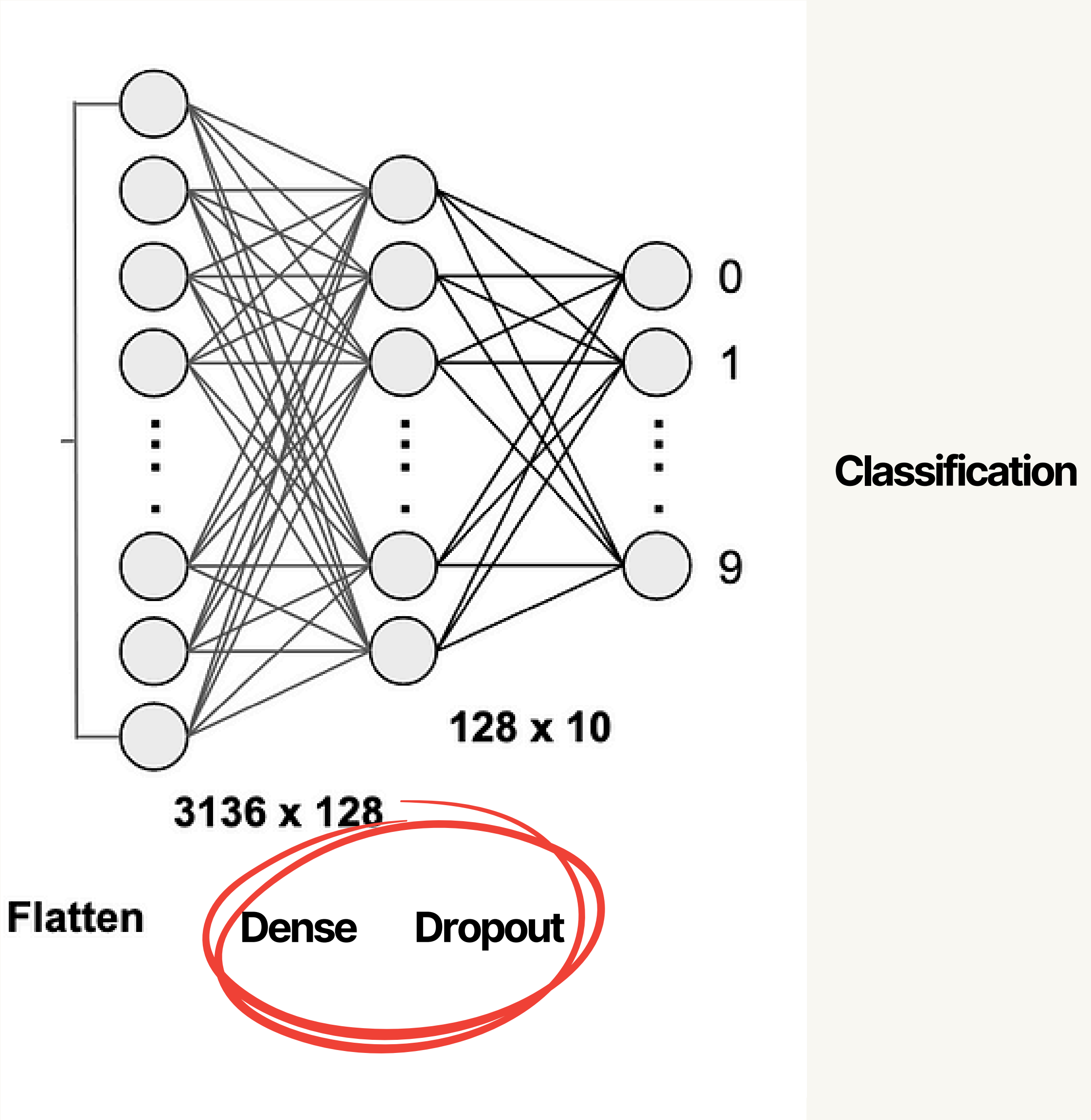


Flatten: Convert 2D tensor to 1D vector – prepare for classification layer.

Fully connect layer - Dense (128 units)
Learn nonlinear, linear relationship between features and output layer.

Dropout layer
Randomly “turn off” some neurons during training.
So other neutrons learn other basic features.

Output layer
Predict the probability of belonging to 47 different character, digit classes (uppercase, lowercase, numbers).



Demo time

04 Strengths and limitations

Strengths:

- **High test Accuracy (~88%)** with a relatively simple CNN architecture.
- **Fast and stable training** using the Adam optimizer and a large batch size (128).
- **Effective character recognition** across digits, uppercase, and lowercase letters.
- **No overfitting issues** observed despite not using heavy regularization techniques

Weaknesses:

- **Confusions between similar characters**, such as:
  - **0** vs **O**
  - **1** vs **l**
  - **q** vs **9**
- Model simplicity **limits its ability to capture very fine variations in handwriting.**
- No data augmentation applied, which reduces robustness to variations in rotation, skew, or writing style.

05 Conclusions

- **CNN is a powerful tool** for **image-based pattern recognition** like handwritten character classification.
- Our model achieved ~88% accuracy, showing solid performance with a clear and compact architecture.
- Common errors arise from visual similarity, which is a natural challenge in handwriting recognition.
- Model performance can be improved by using data augmentation, deeper CNNs, or integrating sequence models (e.g., RNNs or Transformers) for word-level recognition.
- Future work may include combining CNNs with CTC loss or exploring more advanced architectures for better generalization, upgrade the model to word and sentence recognition level

Thank you