# **Letter Recognition – Ideation**

1. **Project Overview**

This project addresses the problem of **detecting anomalous character shapes** in a dataset of handwritten-like characters derived from English capital letters. The dataset contains **20,000 black-and-white pixel images**, summarized into **16 numerical features** such as statistical moments, edge counts, and geometric descriptors.

While conventional approaches focus on classification into 26 categories (A–Z), the emphasis here is on **identifying characters that exhibit unusual distortions or irregular patterns**. Such anomalies, if left undetected, can reduce the accuracy and reliability of Optical Character Recognition (OCR) systems used in **document digitization, postal sorting, and license plate recognition**.

By flagging these anomalous cases, the system can ensure that OCR pipelines maintain high confidence, defer uncertain predictions for human review, and improve **quality control, scalability, and trustworthiness**.

1. **Problem Understanding**

The **core challenge** is not just recognizing characters but ensuring that distorted or rare shapes are detected and separated from normal data.

* **Variability within classes:** Letters from different fonts and random distortions may cause wide variations in feature values, making anomaly detection non-trivial.
* **Overlaps across classes:** Certain letters (e.g., ‘O’ vs ‘Q’ or ‘I’ vs ‘L’) share similar features, and extreme distortions may push them outside their normal cluster.
* **Noise and distortions:** Warping, skewing, or irregular scanning may produce unusual feature patterns that could mislead a classifier if anomalies are not flagged.
* **Objective:** Instead of solely classifying characters, the system must **detect outliers or anomalous shapes** that do not fit well into any class distribution.

In this context, the project is framed as an **anomaly detection task** using techniques such as clustering consistency checks, density-based outlier detection, and statistical deviation analysis.

1. **Data Analysis and Validation**

* Check **shape, duplicates, and missing values**.
* Validate feature ranges to ensure no extreme noise from data entry.
* Inspect the **distribution of values per feature** to detect skewness or long tails (potential anomaly zones).

1. **Exploratory Data Analysis (EDA)**

Unlike classification EDA, the focus here is on **understanding the distribution of “normal” vs “potential anomaly” patterns**:

* **Box plots** for each feature to highlight extreme values.
* **Correlation heatmap** to detect redundant features (e.g., highly correlated variables may bias anomaly detection).
* **Dimensionality reduction (PCA/t-SNE)** to visualize clusters — anomalies often appear as isolated points.
* **Density plots** (KDE) for each feature to see where rare cases occur.

1. **Feature Engineering**

* **Scaling**: Standardize all features to avoid scale dominance.
* **Dimensionality reduction**: Use PCA or autoencoders to compress into fewer latent features where anomalies are more visible.

1. **Model Selection for Anomaly Detection**

Several approaches can be applied:

**Traditional ML Models**

* **Isolation Forest**: Efficient in detecting outliers by random partitioning; works well for high-dimensional data.
* **DBSCAN (unsupervised clustering)**: Naturally isolates noise points that don’t belong to dense clusters.

**Neural Network Models**

* **Autoencoder-based Anomaly Detection**:
  + Train an autoencoder to reconstruct normal characters.
  + Large reconstruction error indicates an anomaly (since the autoencoder fails to learn rare, distorted shapes).
* **Variational Autoencoders (VAE)**: Probabilistic reconstruction models that can quantify uncertainty; anomalies usually have low likelihood.

1. **Training & Evaluation Strategy**

* Train on the **majority of data considered “normal”**, assuming anomalies are rare.
* Evaluate using:
  + **ROC-AUC** (for anomaly scores).
  + **Silhouette / cluster separation scores** (for density-based methods).
* Cross-check detected anomalies with **statistical thresholds** (e.g., z-score > 3) to validate

1. **Error Analysis**

* Inspect flagged anomalies:
  + Are they genuinely distorted characters?
  + Or are they valid but rare variations of certain letters (false positives)?
* Analyze feature patterns of anomalies to discover if certain letters (e.g., ‘I’ or ‘O’) are more prone to misshaping.