



SmartBoard

Un whiteboard intelligent qui comprend ce que vous voulez faire en un seul tracé de souris

Dessinez ce que vous avez en tête, et le whiteboard s'occupera de le rendre réel.

AL AZAWI Rayan

Scénarios d'Utilisation

01 Mindmaps

02 Automates

03 Prise de notes

04 Diagrammes UML

05 Présentation de
cours intelligente

En un seul clic de souris

Un problème complexe

Plusieurs contraintes à respecter

Création de tableau blanc
(pas si facile, Miro \$17.5B)

- Accessible sur tout type d'écran (web app)
- Permettre un tracé à main libre fluide (render loop, state management)
- Créer et gérer des coordonnées (trigo, algèbre vectorielle)
- Représentation d'un tracé (liste de points)
- Représentation d'une forme (dépendant)

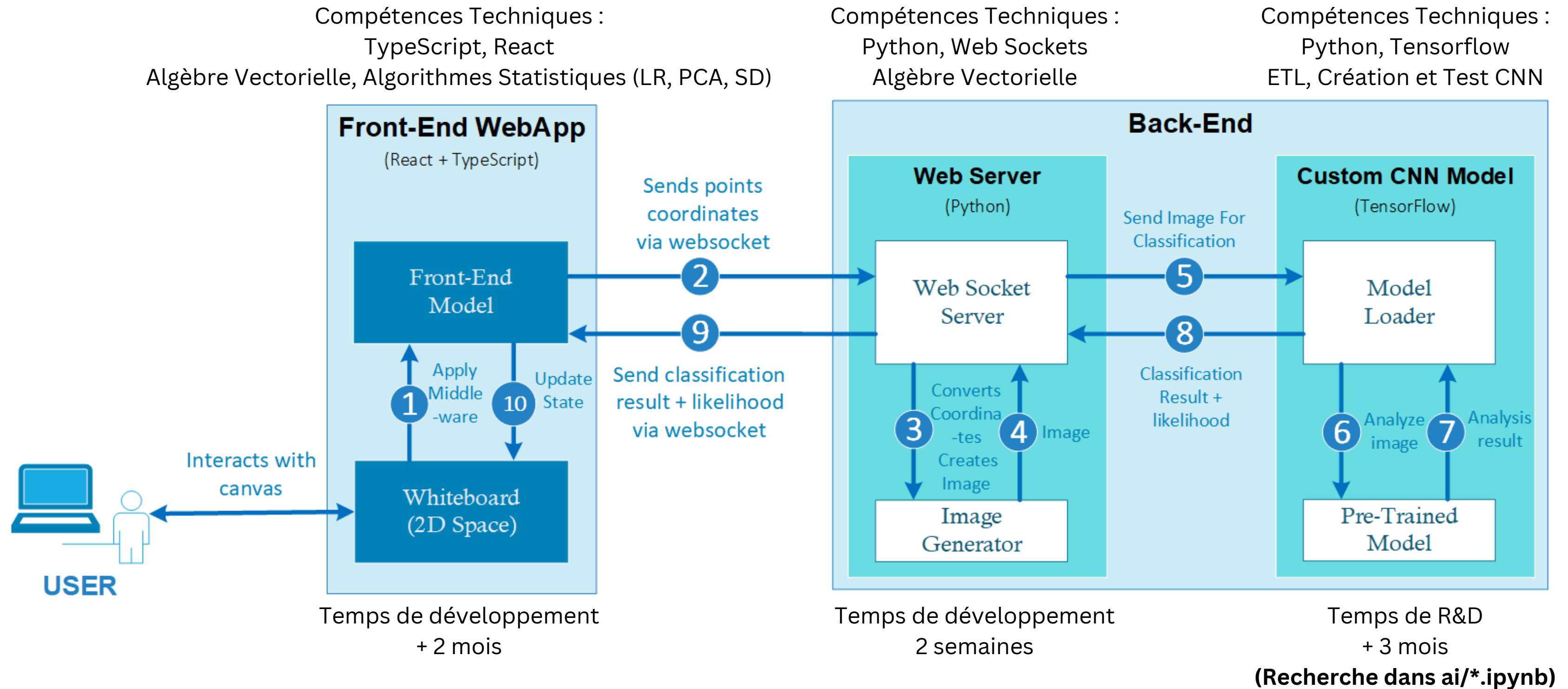
Le rendre intelligent
(Backend et Frontend)

- Recherche, Extract-Transform-Load (ETL) d'un dataset de formes
- Recherche scientifique, modélisation, création et entraînement de dizaines de modèles IA
- Test de performance de chaque modèle, bench du meilleur (sur l'accuracy)
- Fonctionnalités intelligentes (algorithmes) indépendantes du backend

Serveur qui relie les deux
(avec latence minimale)

- Protocole de communications client-serveur
- Classifications quasi instantanées (client-serveur-modèle-client)
- Client envoie le minimum d'informations nécessaires
- Serveur reconstitue l'information complète et demande une classification
- Benchmark du temps de classification

Diagramme d'Architecture Global



Les 3 paquets (frontend, backend, ai) sont statiquement typés, lintés, documentés. Il n'y a pas de magic numbers (consts.ts/py). Ils contiennent des instructions d'installation et de lancement (INSTALLATION.md) dans chaque répertoire.

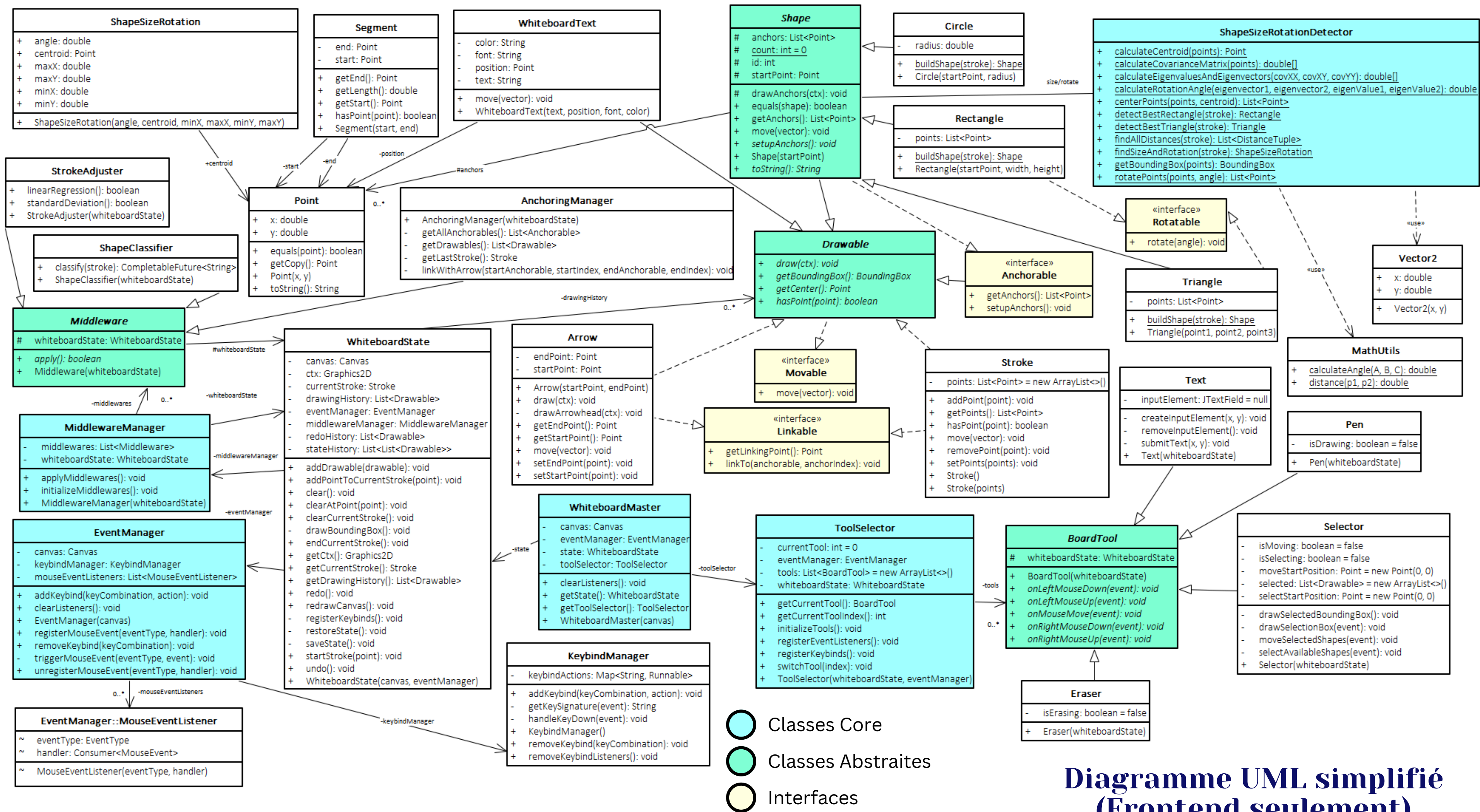


Diagramme UML simplifié
(Frontend seulement)

Testing & Difficultés techniques

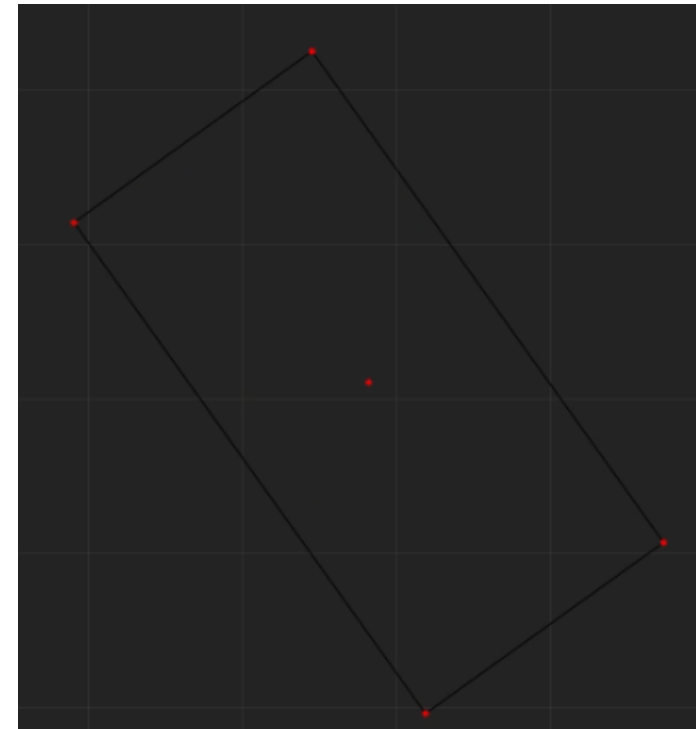
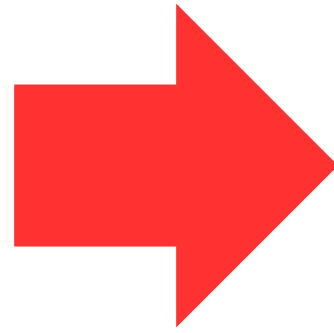
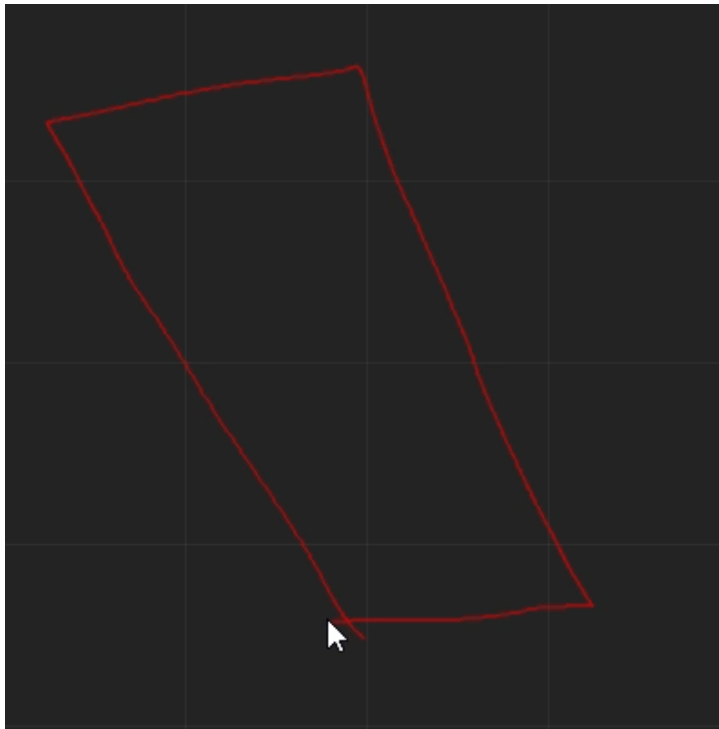
Comment tester un projet pareil :

- Benchmark la précision du modèle sur un échantillon test qu'il n'a jamais vu post training
- Benchmark la vitesse de classification (tout le processus client -> server -> model -> client)
- Tests unitaires sur des fonctions mathématiques et algorithmes (distance cartésienne, calcul d'angle, regression linéaire, écart type, calcul bounding box, algo PCA etc...)
- Test utilisateur sur la précision des formes détectées, notamment sur la correction de la forme (classification modèle), la taille (bounding box) et la rotation (bounding box + PCA)

Quelques difficultés techniques et leurs solutions :

- Classification imprécise sur des tracés autres que le dataset (tracé de souris)
 - CNN, dropout après conv/pooling, data augmentation, learning rate scheduling, cf ipynb
- Trouver la taille et la rotation d'une forme après classification avec une excellente précision
 - Bounding box sur les points + calcul centroïde pour la taille, algorithme BCA pour la rotation (meilleurs résultats que la régression linéaire sur les points du tracé)
- Classification lente, grande latence utilisateur entre le tracé et la détection de forme
 - Websockets, envoi d'une liste de points au lieu d'une image (le modèle fonctionne sur des images), reconstitution serveur d'une image (conversion coordonnées, algo scaling etc...)
- Ajout d'outils (Pen, Selection, Eraser, Text) extensibles dépendant des inputs utilisateurs
 - Création d'un Event Manager global utilisant le patron Observateur, les outils implémentent Tool qui est observateur, création d'un ToolManager pour gérer l'ensemble

Quelques Benchmarks



Benchmark classification

```
Classification took 41.40000000002328ms. ClassifierAPI.ts:19  
Size / Rotation detection took 0.6000000000931323ms. ShapeClassifier.ts?t=1716315053579:31
```

Classification (client -> serveur -> modèle -> client) : 41ms

Calcul taille / rotation (Bounding Box et PCA) : 0.6ms

```
1/1 ————— 0s 13ms/step  
Predicted class: rectangle, Likelihood: 0.996951699256897
```

Classification (modèle): 13ms

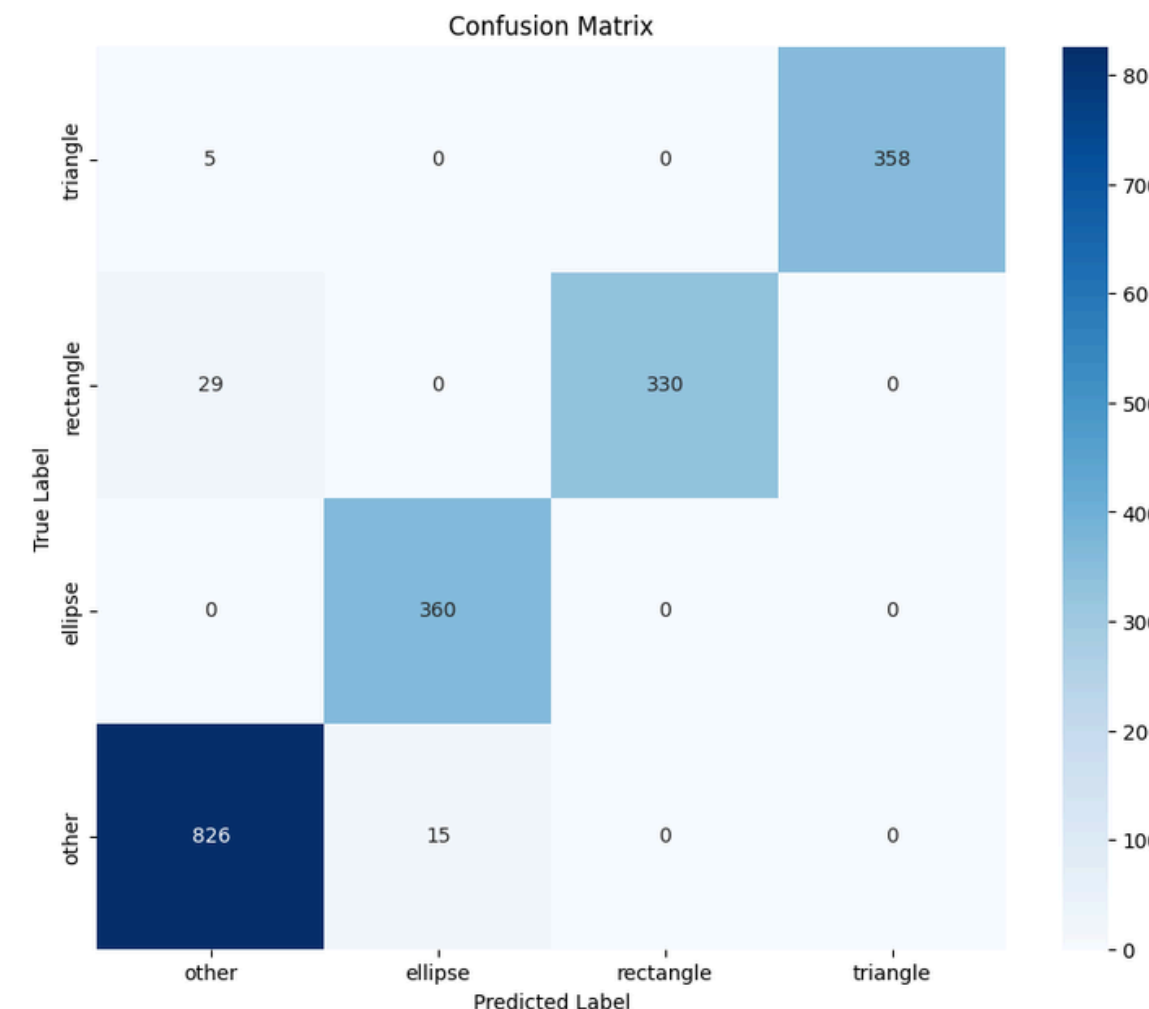
Overhead serveur en local : $41 - 13 = 28\text{ms}$ (très rapide)

Benchmark modèle

```
61/61 ————— 3s 17ms/step  
Test accuracy: 0.9745189547538757
```

Tests sur un échantillon jamais vu :
Accuracy > 97% (très efficace)

Probablement de meilleurs résultats
avec **une meilleure carte graphique**
et plus d'epochs (10 epochs
prenaient +1h d'entraînement)



Tout le processus de recherche, ETL, modélisation, création entraînement et test du modèle est documenté scientifiquement et justifié dans un Python Notebook interactif (dans le répertoire AI, finissant par ipynb).

En l'utilisant (INSTALLATION.md), vous pouvez effectuer tout le processus étape par étape, y compris tester et générer cette matrice de confusion sur votre propre machine.

Programmation

```
static findSizeAndRotation(stroke: Stroke): ShapeSizeRotation {
  const points = stroke.getPoints();
  const centroid = this.calculateCentroid(points);
  const centeredPoints = this.centerPoints(points, centroid);
  const [covXX, covXY, covYY] =
    this.calculateCovarianceMatrix(centeredPoints);
  const [eigenvector1, eigenvector2, eigenValue1, eigenValue2] =
    this.calculateEigenvaluesAndEigenvectors(covXX, covXY, covYY);
  const angle = this.calculateRotationAngle(
    eigenvector1,
    eigenvector2,
    eigenValue1,
    eigenValue2
  );
  const rotatedPoints = this.rotatePoints(centeredPoints, angle);
  const { minX, maxX, minY, maxY } = this.getBoundingBox(rotatedPoints);
  return { angle, centroid, minX, maxX, minY, maxY };
}

static detectBestRectangle(stroke: Stroke): Rectangle {
  const { angle, centroid, minX, maxX, minY, maxY } =
    this.findSizeAndRotation(stroke);

  const width = maxX - minX;
  const height = maxY - minY;
  const rectStartPoint = new Point(minX + centroid.x, minY + centroid.y);
  const rectangle = new Rectangle(rectStartPoint, width, height);
  rectangle.rotate(angle * (180 / Math.PI));
  return rectangle;
}
```

frontend/src/util/ShapeSizeRotationDetector.ts

SmartBoard v2

Features possibles pour la v2

- Chat vocal et écrit (donner un cours à distance)
- Permettre la collaboration à distance avec gestion d'accès et d'interaction par le créateur d'un whiteboard
- Améliorer UI, donner plus d'outils de dessin, permettre l'alignement des formes entre elles
- Ajouter plus de formes détectées au modèle IA