Rapport

Reconnaissance des caractères par inférence active

CENTRALE MARSEILLE

Projet S8 - Science Informatique, Science Numérique

École Centrale de Marseille

38 Rue Joliot Curie

13013 Marseille

Laura FRANKE

Pauline DAME

Samuel CHARTRER

Introduction

Depuis les 10 dernières années, la place du *machine learning* (fr.: apprentissage des machines) dans l'industrie n'a fait que croitre. Le *machine learning* correspond à l'implémentation de méthodes d'intelligence artificielle de telle sorte qu'une machine apprend la tâche qu'elle doit réaliser par elle-même sans qu'un humain ait à programmer un ensemble de cas et de règles. De cette façon, un logiciel peut par exemple apprendre à reconnaître si un morceau de musique est de Bach ou Vivaldi. Avec le temps, différentes méthodes ont été développés. Une méthode très employée est le réseau de neurones convolutifs qui s'inspire des réseaux de neurones de notre cerveau.

Le *machine learning* est un sujet qui a attiré notre attention de par son importance dans l'industrie, la science et notre société. Nous avons voulu comprendre les concepts qui le sous-tendent et à implémenter un programme par nous-même. De plus, nous avons été très intéressés par le principe du projet de s'inspirer du fonctionnement de la vision humaine.

Dans ce rapport, nous détaillerons le but du projet avant d'en détailler l'implémentation puis d'en analyser les résultats.

Descriptif du projet

But du projet

Le but de ce projet est d'implémenter un ou plusieurs algorithmes de reconnaissances de chiffres manuscrits en se basant sur le système visuel humain. En effet, le pré-traitement des donnés se fait par des filtres qui permettent de simuler la vision périphérique, et de plus, lors de la classification, les algorithmes se comportent comme l'oeil en se déplaçant par saccades d'un point d'intérêt à un autre afin d'identifier le chiffre présent sur l'image.

Base de donées

Notre projet exploite la base de données MNIST (Modified National Institute of Standards and Technology database). Celle-ci contient 70.000 images manuscrites de taille 28x28 pixels représentant des chiffres entre 0 et 9, et est divisée en deux parties. La première contient 60.000 images destinées à l'entraînement de l'algorithme, et la deuxième à son test. Les images sont représentées par des matrices contenant la valeur de chaque pixel en niveau de gris. À chaque image est associée le chiffre qu'elle représente par le biais de deux bases de labels.

Pré-traitement et entraînement de l'algorithme

Les algorithmes s'inspirent du fonctionnement de nos yeux. Ceux-ci se fixent sur un point dans l'image qui apparaît net, les points voisins étant de plus en plus flous à mesure que l'on s'éloigne ce point central. L'observation du trajet parcouru par la pupille dans une expérience de reconnaissance d'image montre de plus que le regard se fixe préférentiellement sur les zones présentant les caractéristiques les moins attendues. Ainsi, ces deux phénomènes sont modélisés par l'application de filtres orthogonaux sur l'image. On applique 4 filtres pour détecter les contours verticaux, horizontaux, obliques et un dernier pour moyenner l'image. On utilise 4 tailles de filtre différentes, on les applique centrés sur un point de l'image et on regroupe les

valeurs après filtrage dans un vecteur de taille 16. En centrant les filtres on obtient des informations plus précises pour les pixels proches du point auquel on s'intéresse et des informations moyennées pour les points qui ne se verront appliquer que les filtres de grande taille. C'est ainsi que l'on simule la baisse de résolution observée dans la vision périphérique humaine.

$$f_1 = \begin{pmatrix} 1 & 1 & -1 & -1 \\ 1 & 1 & -1 & -1 \\ 1 & 1 & -1 & -1 \\ 1 & 1 & -1 & -1 \end{pmatrix} \quad f_2 = \begin{pmatrix} 1 & 1 & -1 & -1 \\ 1 & 1 & -1 & -1 \\ -1 & -1 & 1 & 1 \\ -1 & -1 & 1 & 1 \end{pmatrix} \quad f_3 = \begin{pmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \\ -1 & -1 & -1 & -1 \\ -1 & -1 & -1 & -1 \end{pmatrix} \quad f_4 = \begin{pmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \end{pmatrix}$$

Allure des filtres appliqués sur l'image

Avec les résultats obtenus, on détermine une distribution des valeurs des vecteurs de caractéristiques pour chacun des chiffres en fonction de la position du point observé.

Analyse de l'image

Une fois les distributions établies, on choisit un point de l'image pour lequel on compare le vecteur de caractéristiques extrait avec les distributions établies pour chaque chiffre en ce même point afin de formuler une hypothèse sur la valeur du chiffre représenté. On choisit ensuite un second point pour confirmer ou infirmer l'hypothèse, c'est-à-dire actualiser les probabilités que l'on prête à chaque valeur possible. On itère le processus jusqu'à ce que l'une des probabilité dépasse une valeur seuil. Nous avons implémenté trois algorithmes basé sur cette méthode. Ils diffèrent par le choix des positions de l'image utilisées pour réaliser l'analyse.

Nous avons également testé plusieurs valeurs seuils. Les résultats en fonction de ces différentes valeurs sont détaillés dans les parties suivantes et en annexe.

Traitement de la base d'image

Notre algorithme est constitué de deux parties : la création de la base de données et l'implémentation des algorithmes de reconnaissance. En ce qui concerne le format des données, nous avons choisi de travailler en convertissant les matrices correspondant aux images en vecteurs. Nous réalisons ensuite le filtrage en faisant le produit scalaire entre les 16 filtres et le vecteur image pour chaque point.

On obtient ainsi une base de données contenant des élément de type $\{v, y, u\}$ où *v* est le résultat du filtrage à la position u et *y* la classe de l'image. A partir de cette base, on calcule pour chaque classe i la moyenne $\overrightarrow{\mu}^i_u$ des valeurs des vecteurs de caractéristiques en chaque position u, ainsi que la matrice de covariance δ^i_u. Nous en déduisons les distributions de probabilité correspondant à chaque chiffre que nous supposons gaussiennes.

Algorithmes de reconnaissance

Algorithme n°1

Principe:

On filtre l'image pour obtenir les vecteurs de caractéristiques $\overrightarrow{v}_u$ pour toutes les positions u. A chaque tour de boucle, on cherche le point de l'image à analyser le plus surprenant par rapport à l'hypothèse courante jusqu'à avoir une probabilité supérieur au seuil de décision pour un des chiffres.

Initialisation:

S est le seuil de décision, U l'ensemble des points parcourus.

$0 \leq S < 1$

$U = \varnothing$

$\forall i \in [0,9], P(i) = \frac{1}{10}$

$u = argmin_{u' \notin U} \sum_{h=0}^{9} ln(P(h)N(\overrightarrow{v}_{u'}, \overrightarrow{\mu}^h_{u'}, \delta^h_{u'}))$

$\forall i, P'(i) = \frac{P(i)N(\overrightarrow{v_u}, \overrightarrow{\mu}^i_u, \delta^i_u)}{\sum_{h=0}^{9} P(h)N(\overrightarrow{v_u}, \overrightarrow{\mu}^h_u, \delta^h_u)}$

$\forall i, P(i) \leftarrow P'(i)$

$U \leftarrow U + [u]$

Boucle principale:

Tant que $\nexists$ *i tel que* $P(i) > S$:

$H = argmax_{h \in [0;9]} P(h)$

$u = argmin_{u' \notin U} ln(P(H)N(\overrightarrow{v}_{u'}, \overrightarrow{\mu}^H_{u'}, \delta^H_{u'}))$

$\forall i, P'(i) = \frac{P(i)N(\overrightarrow{v_u}, \overrightarrow{\mu}^i_u, \delta^i_u)}{\sum_{h=0}^{9} P(h)N(\overrightarrow{v_u}, \overrightarrow{\mu}^h_u, \delta^h_u)}$

$\forall i, P(i) \leftarrow P'(i)$

$U \leftarrow U + [u]$

$H = argmax_{h \in [0;9]} P(h)$

Algorithme n°2

Principe:

A chaque boucle, on cherche le point le plus susceptible de confirmer l'hypothèse courante en se basant sur les distributions de la base d'entraînement, jusqu'à avoir une probabilité supérieur au seuil de décision pour un des chiffres. On ne calcule $\overrightarrow{v}_u$ que pour les points choisis.

Initialisation:

$0 \leq S < 1$

$U = \varnothing$

$u = \textit{centre de l'image}$

$\forall i, P'(i) = \frac{N(\overrightarrow{v_u}, \overrightarrow{\mu}^i_u, \delta^i_u)}{\sum_{h=0}^{9} N(\overrightarrow{v_u}, \overrightarrow{\mu}^h_u, \delta^h_u)}$

$\forall i, P(i) \leftarrow P'(i)$

Boucle principale:

$\textit{Tant que } \nexists\, i \textit{ tel que } P(i) > S \ :$

$H = argmax_{h \in [0;9]} P(h)$

$u = argmax_{u' \notin U} P(H) N(\overrightarrow{\mu}^H_{u'}, \overrightarrow{\mu}^H_{u'}, \delta^H_{u'})$

$\forall i, P'(i) = \frac{P(i) N(\overrightarrow{v_u}, \overrightarrow{\mu}^i_u, \delta^i_u)}{\sum_{h=0}^{9} P(h) N(\overrightarrow{v_u}, \overrightarrow{\mu}^h_u, \delta^h_u)}$

$\forall i, P(i) \leftarrow P'(i)$

$U \leftarrow U + [u]$

$H = argmax_{h \in [0;9]} P(h)$

Algorithme n°3

Cet algorithme (*Algorithme de reconnaissance prédictive*) est similaire à l'algorithme n°2, à la différence qu'on choisit ici le point le plus susceptible d'infirmer l'hypothèse avec :

$u = argmin_{u' \notin U} P(H) N(\overrightarrow{\mu}^H_{u'}, \overrightarrow{\mu}^H_{u'}, \delta^H_{u'})$

Résultats et analyse

Algorithme	τ	σ	temps par image
1	0.8378	0.0037	3.44s
2	0.6869	0.0046	2.19s
3	0.8987	0.0030	3.64s

Résultats sur 10000 images avec un seuil de 0.999

Nous avons constaté que le taux de réussite chute pour l'algorithme 2 lorsque le seuil de probabilité pour la prise de décision augment. Il semble donc que la recherche du point d'intérêt suivant pour l'algorithme n°2 pose problème, c'est ce qui a motivé la création de l'algorithme n°3. En changeant le argmax en argmin dans cette recherche pour l'algorithme n°3 le problème n'apparait plus.

Chercher à infirmer l'hypothèse courante donne le meilleur résultat, en effet en cherchant à la confirmer on peut se restreindre à analyser des points communs à plusieurs chiffres et ne pas savoir les différencier. Ainsi en analysant les matrices de confusion où l'algorithme une autre chiffre comme étant et certains exemples, nous avons constaté que l'algorithme n°2 n'arrive pas à identifier les 4 car dans la plupart des cas ils ne se différencient des 9 que par le haut de l'image.

D'un point de vu biologique cela semble correspondre au fait que le regard est davantage attiré par ce qu'on ne s'attend pas à voir.

Les plus grands filtres appliqués sont presque aussi grands que la plus petite surface contenant les chiffres, en prenant un point au milieu de l'image on a donc déjà beaucoup d'informations sur l'image, ce qui pourrait expliquer les 82% de réussite observés lorsque l'on ne s'intéresse qu'au point central.

La différence de temps de calculs entre les algorithmes 1 et 3 n'est pas significative (5%). Cependant n'utiliser que le point centrale de l'algorithme 2 donne des résultats proches à l'algorithme 1 en étant 430 fois plus rapide.

Problèmes rencontrés

Notre premier problème a été la création de la base des vecteurs de caractéristiques et des distributions. La base MNIST étant volumineuse et le nombre de points auxquels effectuer le filtrage élevé, nos ordinateurs n'étaient pas capable d'effectuer les calculs en une fois. Nous avons donc découpé la base originale en dix parties et avons fait les calculs sur dix ordinateurs en parallèle.

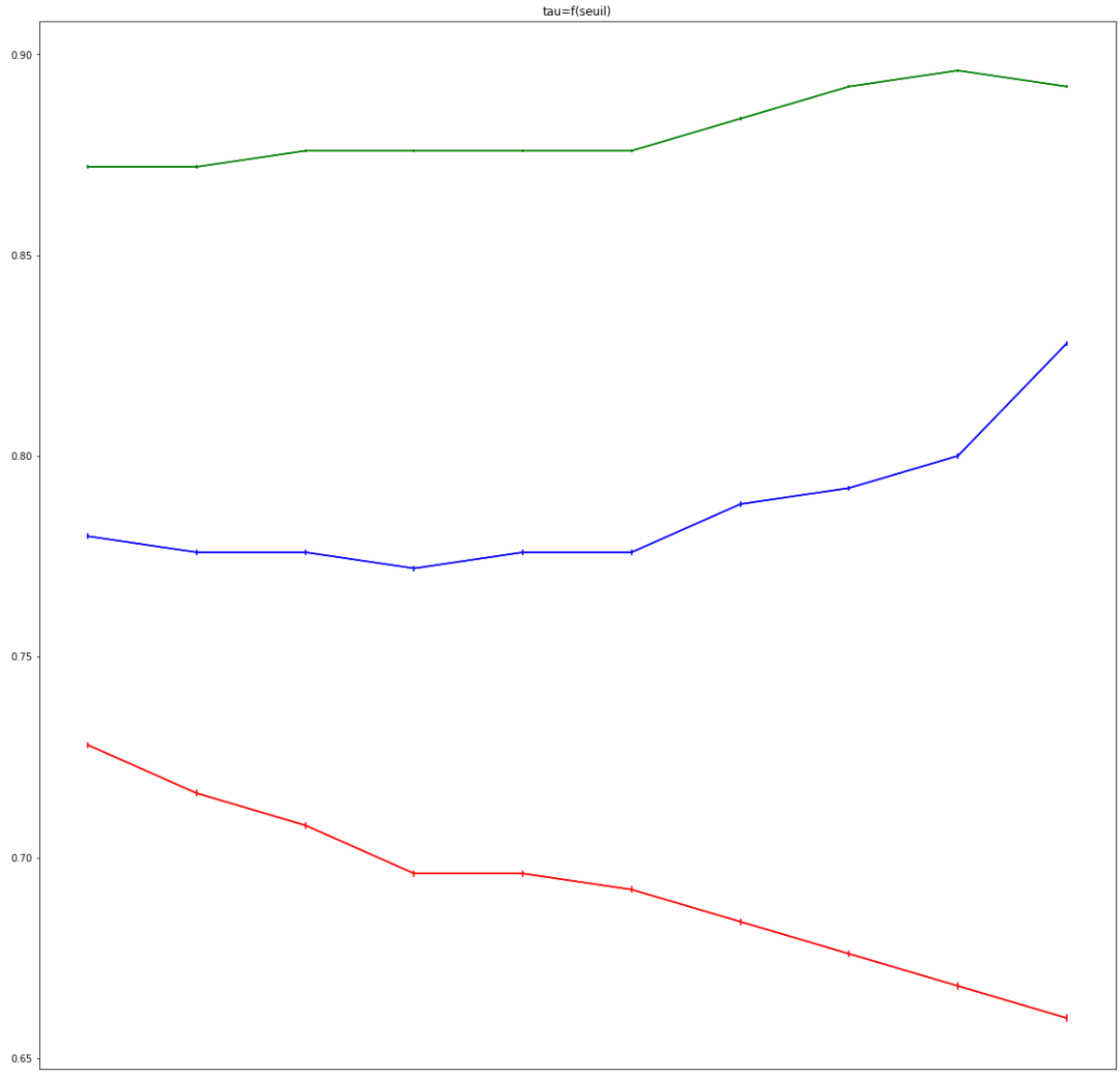
Malheureusement, il existe dans la base des matrices de covariance non-inversible (due à des valeurs nulles ou valeurs corrélées). Il est donc impossible de définir les densités de probabilité en ces points. Pour pouvoir parcourir l'image sans avoir d'erreur en arrivant sur un de ces points il a donc fallu les recenser et les exclures des points à parcourir. Heureusement ces points sont situés sur les bords des images où il n'y a en général rien.

Conclusion

L'idée intrigante d'implémenter un algorithmes de reconnaissances de chiffres manuscrits en se basant sur le système visuel humain a un taux de réussite étonnamment élevé, de presque 90% pour les algorithmes n°1 et n°3. Cependant, en comparaison avec des autres modèles établis, où un taux de réussite de 95% est le minimum pour valider un classificateur, notre méthode n'est pas la meilleure, et cela est peut-être dû aux principes même de la méthode. Une hypothèse est que le modèle mathématique choisi pour les distributions n'est pas le bon. D'autres filtres pourraient donner de meilleurs résultats. Une expérience envisageable serait de mélanger les algorithmes n°1 et n°3 afin de chercher les points d'intérêt aussi bien dans l'image analysée que dans la représentation que l'on a des chiffres.

Le résultat de chercher à infirmer l'hypothèse courante donne le meilleur solution et ce résultat est compréhensible lorsqu'on prend en compte que l'oeil ne se fixe non pas uniquement sur les similarités entre le chiffre présenté et sa connaissance des formes de chiffres, mais également sur ce qu'on ne s'attend pas à voir.

Annexes



Résultats en fonction du seuil (échelle log, 0.95<seuil<0.995)

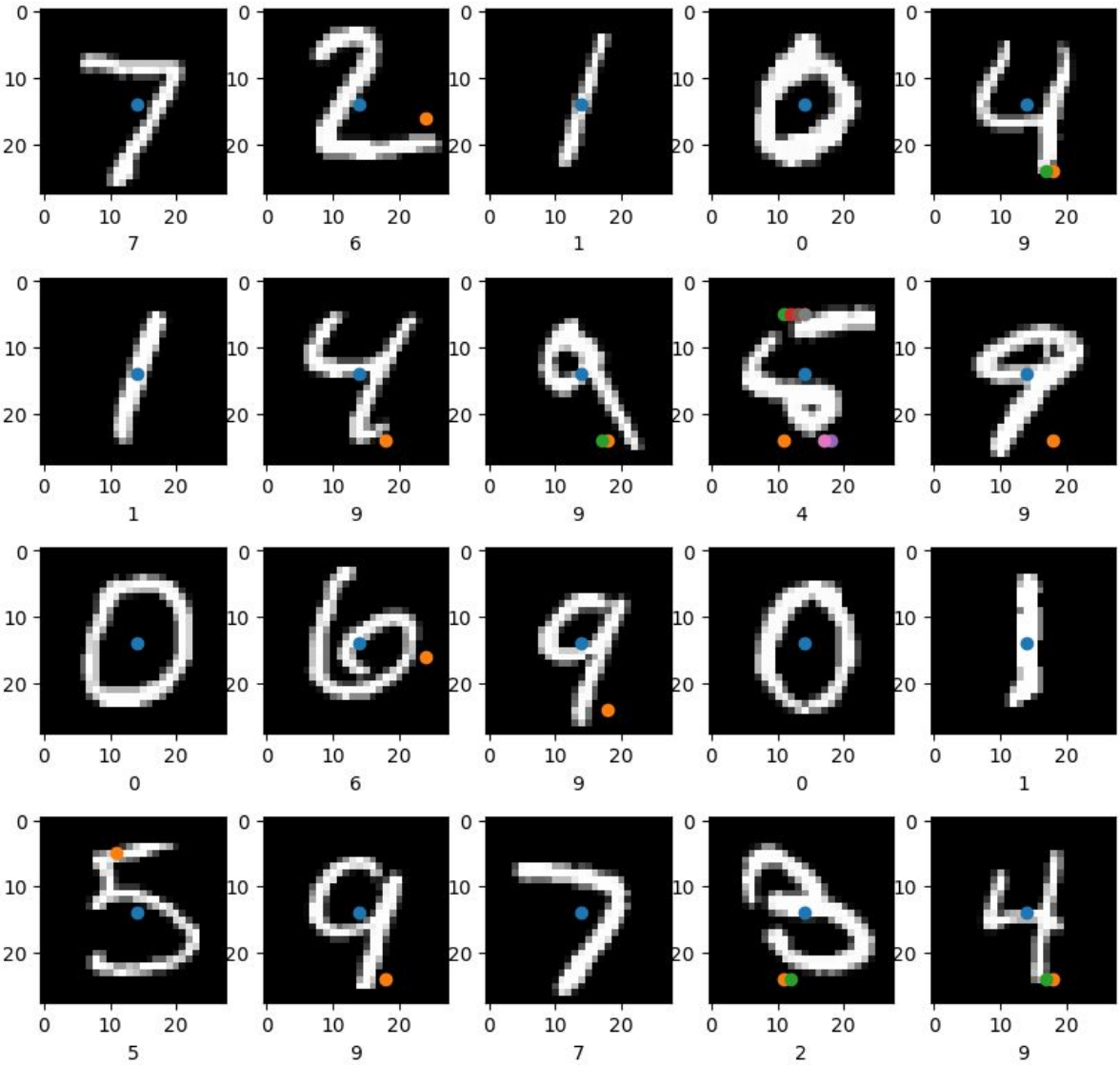
1	0.7237	0.0044	2.5s
2 & 3	0.8277	0.0037	0.008s

Résultats en ne prenant que le 1er point:

Points parcourus par algorithme n°1 lors de la classifications de 20 images de la base de test

classe estimée \ classe réelle	0	1	2	3	4	5	6	7	8	9
0	0.93	0.	0.	0.	0.	0.04	0.04	0.	0.03	0.01
1	0.	0.9	0.	0.	0.	0.01	0.	0.	0.01	0.
2	0.03	0.03	0.93	0.11	0.1	0.03	0.03	0.07	0.07	0.03
3	0.	0.03	0.	0.79	0.	0.06	0.	0.04	0.07	0.
4	0.	0.02	0.01	0.01	0.72	0.	0.01	0.09	0.02	0.05
5	0.02	0.	0.	0.02	0.01	0.75	0.03	0.01	0.03	0.
6	0.01	0.	0.01	0.01	0.04	0.01	0.87	0.	0.	0.
7	0.	0.	0.01	0.02	0.06	0.01	0.	0.64	0.01	0.02
8	0.01	0.02	0.04	0.03	0.05	0.08	0.02	0.07	0.76	0.06
9	0.	0.	0.	0.01	0.02	0.01	0.	0.08	0.	0.83

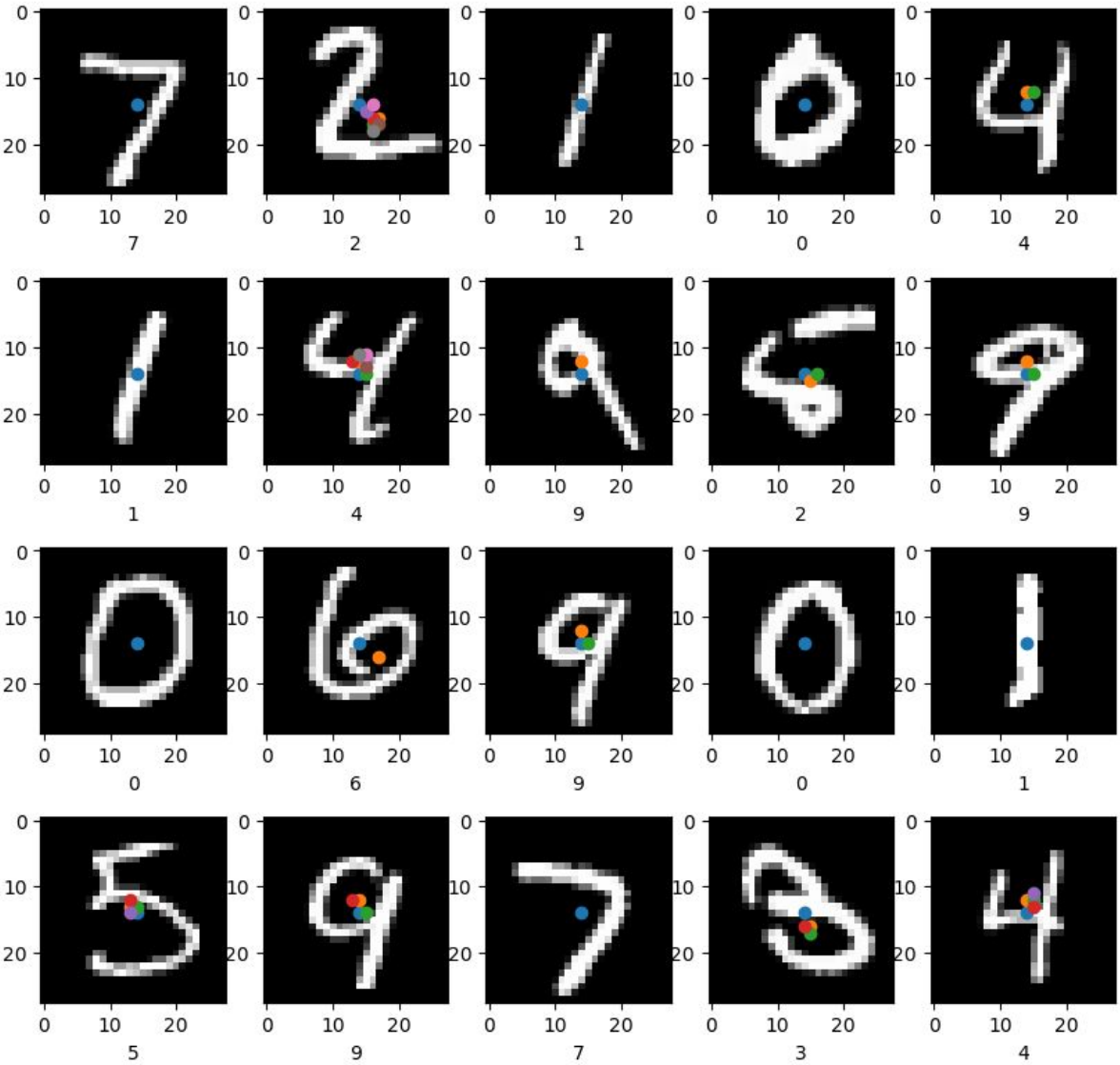
Matrice de confusion de l'algorithme 1



Points parcourus par algorithme n°2 lors de la classifications de 20 images de la base de test

classe estimée \ classe réelle	0	1	2	3	4	5	6	7	8	9
0	0.88	0.	0.	0.	0.	0.	0.01	0.01	0.	0.
1	0.01	0.99	0.1	0.09	0.01	0.1	0.	0.07	0.15	0.01
2	0.	0.	0.39	0.01	0.01	0.	0.	0.	0.	0.
3	0.	0.	0.33	0.77	0.2	0.08	0.03	0.05	0.06	0.06
4	0.	0.	0.	0.	0.02	0.01	0.	0.04	0.01	0.01
5	0.	0.	0.	0.04	0.	0.36	0.01	0.01	0.02	0.
6	0.06	0.	0.06	0.02	0.01	0.11	0.93	0.01	0.07	0.
7	0.01	0.	0.03	0.	0.	0.	0.	0.64	0.01	0.03
8	0.04	0.01	0.02	0.03	0.04	0.23	0.	0.04	0.59	0.06
9	0.	0.	0.07	0.04	0.71	0.11	0.02	0.13	0.09	0.83

Matrice de confusion de l'algorithme 2



Points parcourus par algorithme n°3 lors de la classifications de 20 images de la base de test

classe estimée \ classe réelle	0	1	2	3	4	5	6	7	8	9
0	0.96	0.	0.02	0.	0.	0.	0.02	0.02	0.	0.
1	0.	0.97	0.01	0.	0.	0.	0.	0.01	0.	0.01
2	0.	0.	0.87	0.02	0.01	0.01	0.03	0.13	0.	0.
3	0.01	0.01	0.01	0.82	0.01	0.05	0.	0.	0.06	0.01
4	0.	0.	0.02	0.	0.86	0.03	0.02	0.02	0.	0.05
5	0.01	0.	0.	0.12	0.02	0.84	0.09	0.01	0.06	0.
6	0.	0.	0.01	0.01	0.01	0.02	0.84	0.	0.01	0.
7	0.02	0.	0.03	0.01	0.	0.	0.	0.73	0.01	0.01
8	0.	0.02	0.03	0.	0.03	0.03	0.	0.01	0.84	0.03
9	0.	0.	0.	0.02	0.06	0.02	0.	0.07	0.02	0.89

Matrice de confusion de l'algorithme 3