***Article : Classement des images en utilisant l’algorithme***

***K-NN***

***Abstract :***

*D*

*ans ce projet le travail implémenté est un algorithme de classement des images par une méthode de classement supervisé en utilisant le KNN (K Nearest Neighbors ou K plus proches voisins), En passant par des étapes précises commençant par la détection de contour, le suivie de contour, le reparamétrage des images et le calcul de l’invariant avant passer à la dernière étape qui résume le travail, le calcul de la distance pour classifier les images en se basant sue les invariants calculés.*

***Introduction :***

*Le Machine Learning ou apprentissage automatique est un domaine scientifique. Elle consiste à laisser des algorithmes découvrir des ” patterns “, à savoir des motifs récurrents, dans les ensembles de données. Ces données peuvent être des chiffres, des mots, des images, des statistiques…*

*Tout ce qui peut être stocker numériquement peut servir de données pour le Machine Learning. En décelant les patterns dans ces données, les algorithmes apprennent et améliorent leurs performances dans l’exécution d’une tâche spécifique.*

*Citons dans notre cas l’algorithme des K plus proches voisins ou K-Nearest Neighbors (kNN) est un algorithme de Machine Learning qui appartient à la classe des algorithmes d’apprentissage supervisé (un algorithme reçoit un ensemble de données qui est étiqueté avec des valeurs de sorties correspondantes sur lequel il va pouvoir s’entraîner et définir un modèle de prédiction), simple et facile à mettre en œuvre qui peut être utilisé pour résoudre les problèmes de classification et de régression.*

*La différence fondamentale entre le classificateur K-NN et le classificateur Naive Bayes est que, le premier est un classificateur discriminatoire, mais le second est un classificateur génératif.*

*Pour entrer dans les détails, classificateur K-NN est un classificateur paresseux supervisé qui a heuristiques locales. Étant un classificateur paresseux, il est difficile de l’utiliser pour la prédiction en temps réel. Les limites de décision que vous atteignez avec K-NN sont beaucoup plus complexes que n’importe quel arbre de décision, obtenant ainsi une belle classification.*

*Lorsque vous résolvez un problème qui se concentre directement sur la recherche de similitude entre les observations, K-NN fait mieux en raison de sa nature inhérente à optimiser localement. C’est aussi un retournement parce que les valeurs aberrantes peuvent considérablement tuer la performance.*

*De plus, K-NN est le plus susceptible de déborder, et par conséquent, le fait d’ajuster le « k » pour maximiser le rendement de l’ensemble de tests est la voie à suivre. Comme la complexité de l’espace augmente, la précision de K-NN descend et vous auriez besoin de plus de données, mais l’ordre de ce classificateur est n 2 et il devient trop lent.*

*Ainsi, une technique de réduction de dimensionnalité comme PCA, SVD etc sont généralement appliquées et par la suite ce classificateur est utilisé.*

*Naive Bayes est un classificateur avide d’apprentissage et il est beaucoup plus rapide que K-NN. Ainsi, il pourrait être utilisé pour la prédiction en temps réel. Typiquement, le filtrage de spam par email utilise le classificateur Naive Bayes. Il prend une voie d’estimation probabiliste et génère des probabilités pour chaque classe.*

*Il suppose une indépendance conditionnelle entre les caractéristiques et utilise une hypothèse de probabilité maximale. La meilleure partie avec ce classificateur est que, il apprend au fil du temps.*

*Dans une tâche de filtrage de spam, le type de mots de spam dans l’email évolue au fil du temps. De la même façon, le classificateur calcule également les estimations de probabilité pour les nouveaux mots indésirables dans un modèle de « sac de mots » et s’assure qu’il fonctionne bien.*

*Cette caractéristique du classificateur est due à la nature inhérente de l’algorithme étant générative et non discriminatoire.*

***Formulation :***

*On va discuter les étapes d’implémentation de l’algorithme kNN une par une, on commençant par le traitement et la préparation des images,*

*Data Set utilisée : MPEG7,* *L’ensemble de données MPEG7 est un ensemble de données standard et couramment utilisé pour évaluer la correspondance et la classification des formes.*

|  |  |  |  |
| --- | --- | --- | --- |
|  |  |  |  |
|  |  |  |  |

*La détection de contour par le filtre de Canny :*

*Le filtre de Canny (ou détecteur de Canny) est utilisé en traitement d'images pour la détection des contours.*

*L'algorithme a été conçu par John Canny en 1986 pour être optimal suivant trois critères clairement explicités :*

* *bonne détection : faible taux d'erreur dans la signalisation des contours,*
* *bonne localisation : minimisation des distances entre les contours détectés et les contours réels,*
* *clarté de la réponse : une seule réponse par contour et pas de faux positifs.*

*Première étape :*

*Le rehaussement de Canny La première étape de l’algorithme peut se décomposer en deux sous parties. Il s’agit tout d’abord de lisser l’image, afin d’en retirer les impuretés, et de calculer dans un second temps la norme du gradient et l’angle de la normale au gradient pour chaque pixel de l’image lissée.*

*Deuxième étape :*

*Suppression des non-maximums, La suppression des non-maximums est une étape assez rapide. Il s’agit de supprimer de la norme du gradient toutes les valeurs «faibles», les valeurs qui ne sont pas les plus grandes en suivant l’angle de leur gradient.*

*Troisième étape :*

*Seuillage par hystérésis, Le seuillage par hystérésis était certainement la partie du projet le plus complexe à mettre en œuvre. Il s’agit ici d’affiner le filtrage des contours faibles et de garder uniquement les contours significatifs, en utilisant deux seuils.*

*Dernière étape :*

*Trouver le seuil automatiquement lors de la dernière partie du projet, l’objectif était d’améliorer notre fonction hystérésis, et de trouver le seuil Th automatiquement, en analysant le module du gradient Es. Pour calculer Th, il faut calculer la fonction de répartition du module du gradient de l’image, et trouver la valeur pour laquelle la fonction de répartition est inférieure à ph, avec 70% < ph < 95%.*

*Une autre méthode de décoction de contour on peut utiliser le Laplacien .* *Les opérateurs de gradient exploitent le fait qu'un contour dans une image correspond au maximum du gradient dans la direction orthogonale au contour. Or le passage par zéro de la dérivée seconde d'une rupture d'intensité permet également de mettre en évidence le contour. La dérivée seconde est donc déterminée par le calcul du Laplacien*

*Suivie de contour avec l’algorithme de Moore :*

*Le voisinage Moore d’un pixel, P, est l’ensemble de 8 pixels qui partagent un sommet ou un bord avec ce pixel. Ces pixels sont les pixels P1, P2, P3, P4, P5, P6, P7 et P8 illustrés à la figure 1 ci-dessous. Le quartier de Moore (également connu sous le nom de 8-voisins ou voisins indirects) est un concept important qui se pose fréquemment dans la littérature.*

*La principale faiblesse du traçage Moore-Neighbor réside dans le choix du critère d’arrêt, en d’autres termes, quand l’algorithme se termine-t-il ?*

*Dans la description originale de l’algorithme utilisé dans le traçage Moore-Neighbor, le critère d’arrêt est de visiter le pixel de départ pour une deuxième fois. Comme dans le cas de l’algorithme Square Tracing, il s’avère que Moore-Neighbor tracing ne parviendra pas à tracer le contour d’une grande famille de modèles si cela dépendait de ce critère.*

*Algorithme :*

*Définissez M(a) comme le voisinage de Moore du pixel a.*

*Soit p le pixel limite courant.*

*Soit c le pixel courant considéré, c'est-à-dire que c est dans M(p).*

|  |  |
| --- | --- |
| *Begin*   * *Set****B****to be empty.* * *From bottom to top and left to right scan the cells of****T****until a black pixel,****s,****of****P****is found.* * *Insert****s****in****B****.* * *Set the current boundary point****p****to****s****i.e****. p=s*** * *Backtrack i.e. move to the pixel from which****s****was entered.* * *Set****c****to be the next clockwise pixel in****M(p)****.* * *While****c****not equal to****s****do*   *If****c****is black*   * *insert****c****in****B*** * *set****p=c*** * *backtrack (move the current pixel****c****to the pixel from which****p****was entered)*   *else*   * *advance the current pixel****c****to the next clockwise pixel in****M(p)***   *end While*  *End* | *figure-1*    *figure-2* |

*Reparamétrage :*

*Chaque courbe ayant une infinité des équations qui la représente, on a choisie de faire le reparamétrage par abscisse euclidien pour représenter chaque image, de sorte que tous les images soient paramétrées de la même façon.*

*Exemple de reparamétrage :*

|  |  |
| --- | --- |
|  |  |
|  |  |

*Calcul de l’invariant :*

*Les invariants sont une forme des descripteurs, qui représentent des informations le plus descriptif d’une image pour la différentier avec une autre image, alors pour qu’une image soit invariant par rapport à la translation, la rotation, l’échelle et l’homothétie.*

*Dans ce travail les descripteurs de Fourier sont choisis pour le calcul d’invariant.*

*Les descripteurs de Fourrier sont les coefficients de la transformée de Fourrier => les coordonnées dans la base.  
On utilise généralement les descripteurs de Fourrier pour représenter le contour d'un objet. On définit la courbe de contour sous forme paramétrique:  
  
C(t) = X(t)+ i.Y(t)  
  
\* (X(t), Y(t)) représentent les coordonnées des points du contour dans l'image  
\* t est un paramètre donnant la position sur la courbe  
\* la courbe est fermée => la fonction est périodique.   
Calculons alors la transformée de Fourrier (1D, complexe) de C(t), et gardons les premiers coefficients => on obtient les descripteurs de Fourrier du contour.*

*On utilisant ces descripteurs on calcul l’invariant de chaque image par la formule suivante :*

|  |  |
| --- | --- |
| *Cette famille donne des résultats non stables, on a remarqué que les invariants ont une différence remarquable d’une image a une autre malgré qu’elles sont de même classe.* | *Cette famille donne des résultats stables, on a remarqué que les invariants sont proche d’une image a une autre lorsqu’elles sont de même classe* |
| *Cette famille donne des résultats stables et complètes, on a remarqué que les invariants sont très proche d’une image a une autre lorsqu’elles sont de même classe =>meilleur résultat* |

*Application de KNN et calcul de la distance :*

*Cette étape est la dernière, dont on calcul la distance entre les invariants de chaque image de data "test" avec les invariants de chaque image de data "train" et on classifie selon les résultats obtenue.*

*Augmentation de data :*

*Dans cet article, on va parler de l’augmentation des images qui est une méthode très connue en Machine Learning qui permet d’enrichir notre set de données d’apprentissage. les techniques d’augmentations d’images ;*

*Translation (Shift) :*

*La translation consiste à déplacer l’image sur l’axe des X et/ou Y. Le shift horizontal translate l’image vers la gauche ou la droite tandis que le shift vertical translate l’image vers le haut ou le bas.*

*Rotation :*

*La rotation de l’image selon des angles plus fins modifie également la taille finale de l’image.*

*Scale :*

*L’image peut être mise à l’échelle vers l’extérieur (outward) ou vers l’intérieur (inward). Lors de la mise à l’échelle vers l’extérieur, la taille finale de l’image sera plus grande que la taille d’origine.*

*L’application de KNN après l’augmentation de data nous permet de constater que notre algorithme est stable.*

***Conclusion :***

*En théorie, nous aimerions toujours prédire les réponses qualitatives à l'aide du classificateur de Bayes. Mais pour des données réelles, nous ne connaissons pas la distribution conditionnelle de Y étant donné X, et donc le calcul du classificateur de Bayes est impossible. Par conséquent, le classificateur de Bayes sert d'étalon-or inaccessible auquel comparer d'autres méthodes. De nombreuses approches tentent d'estimer la distribution conditionnelle de Y étant donné X, puis de classer une observation donnée dans la classe avec la probabilité estimée la plus élevée. L'une de ces méthodes est le classificateur K-plus proches voisins (KNN).*