

Conception et spécification

1. Conception de Retro2014

1) Plugin

A. DatabaseObjects

I. Classe **APattern** : héritage « IDisposable », « ICloneable »

Propriétés : APattern a une liste de Signatures et un Bitmap image(peut-être).

Méthodes : elle peut réaliser des fonctions :

- 1) Constructeur par « Agora Alto id » et « Agora output path »
- 2) Constructeur par « Agora Alto id », « Agora output path » et une liste de signatures.
- 3) Ajouter une ASignature.
- 4) Sauvegarder une ASignature.
- 5) Charger une ASignature.
- 6) Calculer la distance Euclidian à un autre APattern, « EuclidianDistance(APattern pattern2) ».
- 7) Calculer la distance Euclidian à un autre APattern sur une signature spéciale « EuclidianDistance(APattern pattern2, int indexSignature) »
- 8) Obtenir des informations sur APattern, mais c'est une fonction abstrct.
- 9) Libérer les mémoires inutiles.
- 10) Dupliquer un APattern en utilisant ICloneable, « Clone() ».
- 11) Diviser toutes les signatures par un entier. « DivideSignatures(int p) »
- 12) Supprimer la dernière signature dans la liste.
- 13) Property changed ??

Utilisation hypothèse:

II. Classe **ASignature** : héritage « IDisposable », « ICloneable »

Propriétés : ASignature dépose un vecteur qui représente des attributs d'un APattern

Méthodes : elle peut réaliser des fonctions :

- 1) Surcharger l'opérateur « / »
- 2) Surcharger l'opérateur « + »
- 3) Nommer des features (abstract)
- 4) Récupérer des features sans normé(abstract)
- 5) Plus des features d'une signature à une valeur actuelle.
- 6) Ajouter un feature à signature
- 7) Calculer la distance euclidienne
- 8) Récupérer le nom de signature

III. Classe **Cluster** : une classe important, un stockage des informations de clusters

Propriétés :

- 1) Une identité :m_id
- 2) une liste de APattern
- 3) le path de fichier qui stocke le cluster
- 4) une valeur stocke l'état que si cluster est chargé
- 5) une valeur stocke l'état que si cluster est étiqueté
- 6) nombre de APattern
- 7) une liste de représentants de cluster, une liste de APattern
- 8) ajouter un représentant(APattern) au cluster
- 9) un cluster a une liste de l'étiquette, une liste de méthode et une liste de confiant taux de l'étiquette ?.

Méthodes :

- 1) Constructeur par « nom de cluster fichier » et « path pour déposer cluster »
- 2) Constructeur par « path pour déposer cluster »
- 3) Sauvegarder un cluster dans XML
- 4) Mise à jour la liste de APatterns qui déjà existe dans le fichier XML
- 5) Charger la liste de APartterns de fichier XML
- 6) Ajouter un étiquette pour cluster
- 7) Mise à jour le nouveau étiquette dans le fichier XML
- 8) Supprimer les listes du cluster : la liste de APatterns, la liste de Representatives, la liste de l'étiquettes
- 9) Récupérer le centre APattern(nouveau) de cluster
- 10) Récupérer le rayon du cluster : la distance moyenne de centre APattern à l'autres
- 11) Récupérer le diamètre de cluster : la distance moyenne entre les APatterns
- 12) Ajouter un APattern au cluster
- 13) Ajouter une liste de APatterns
- 14) Supprimer un APattern dans le cluster
- 15) Mise en ordre de la liste de APatterns faisant référence à la distance à le centre APattern
- 16) Property changed

Utilisation hypothèse:

IV. Classe **Database** : stockage des informations de processus

Propriétés :

- 1) Une liste de documents(images) contenant des APatterns
- 2) Une liste de moyen de features de signature
- 3) Une liste de variance de données
- 4) Une liste de variance de données pour chaque signature dont le APattern
- 5) Deux listes de chaque signature : Max valeur et Min valeur (utilisant dans MinMax normalization)
- 6) Nombre de APatterns : totale nombre pour traitement

Méthodes :

- 1) Ajouter un document
- 2) Normmer des données de dataset par la méthode Z-Score
- 3) Normmer des données de dataset par la méthode MinMax
- 4) Mise à jour des données lorsqu'il ajoute un nouveau APattern dans dataset

V. Classe **Document** : un document (image) qui contient un semble de APatterns à traiter

Propriétés :

- 1) Un Database au lequel le document appartient
- 2) Une liste de APatterns
- 3) Un stockage chemin de document

Méthodes :

- 1) Ajouter un APattern au document

VI. Classe **ShapeEoC** : héritage « APattern », elle représente une section de document image

Propriétés :

- 1) Un path de l'original image(un document).
- 2) Des coordonnées de postions de EoC: horizontale, verticale
- 3) La hauteur et largeur de EoC

Méthodes :

- 1) Charger EoC par des pordonnées de positions et la hauteur et la largeur via le path (m_pathToFullImage).
- 2) Obtenir des informations de EoC : Alto Id, des positions, la hauteur et la largeur
- 3) Obtenir le contexte de EoC
- 4) Constructeur par AgoraAlto id et le path de répertoire de Alto hérite la méthode de APattern
- 5) Constructeur par AgoraAlto id, le path de répertoire de Alto et une liste de signatures hérite la méthode de APattern
- 6) Constructeur avec des informations de EoC, comme des positions, largeur, hauteur)..
- 7) Dupliquer un EoC : Clone()
- 8) Ajouter les signatures d'autre APattern au APattern actuel

Utilisation hypothèse:

B. Interfaces

I. Classe **IClusteringPlugin** : les méthodes pour le processus Clustering

Méthodes :

- 1) Récupérer une liste de clusters par
- 2) Récupérer une liste de clusters par

- 3) Définir le path de clusters dossier
- 4) Définir le database à analyser
- 5) Récupérer le nom de plugin
- 6) Récupérer le fenêtre de configuration du plugin
- 7) Récupérer les informations de plugin(information)
- 8) Récupérer la classe intégrant la configuration du plugin
- 9) Récupérer le temps de processus

II. Classe **IConfig** : pour intégrer le plugin configuraiton

Méthodes :

- 1) Sauvegarder la configuration(le path) et la sérialise à XML
- 2) Dé-sérialise via le path gardé la sérialisation

III. Classe **IDescriptorPlugin**

Méthodes :

- 1) Calculer et Append un signature à un APattern
- 2) Récupérer des information de plugin ou descripteur (configuration)
- 3) Récupérer le nom de descripteur plugin
- 4) Récupérer l'auteur et la configuration du plugin
- 5) Récupérer la fenêtre de configuration.

IV. Classe **IDocumentReaderPlugin** : lecture de document plugins

Méthodes :

- 1) Charger le Database dans le mémoire via la liste de descripteur et le path de données entrées
- 2) Charger le Database via la liste de descripteur, le path de données entrées et la fenêtre principale pour affichage
- 3) Récupérer le temps de charge le dataset
- 4) Récupérer le dataset pendant le processus
- 5) Récupérer la fenêtre configuration
- 6) Récupérer le plugin configuration
- 7) Récupérer l'auteur et nom de Document lecture plugin

C. PluginTools

- I. Classe **BoundingBoxNoiseRemoval** : enlève des bruit de bord de une section de l'image (bounding box)
- II. Classe **ImageNormalisationTools** : le traitement de l'images , mais les techniques ???

2) RetroCore

A. Model

I. Classe **RetroProject** : la modèle de Retro projet

Propriétés :

- 1) Nom de projet Retro
- 2) Le path sauvegardé le projet Retro
- 3) Le path sauvegardé le projet Agora
- 4) Le path sauvegardé l'Alto xml fichier
- 5) Le path sauvegardé l'Agora images fichier
- 6) Le path sauvegardé le résultat de Retro Clustering
- 7) Une liste de clusters
- 8) Le nombre de clusters
- 9) Le total nombre de forme dont les clusters dans le projet

Méthodes :

- 1) Création, ouverture et sauvegarde le projet Retro
- 2) Vide la liste de clusters de projet actuel
- 3) Exporter comme Alto
- 4) Charge la liste de clusters

II. Classe **ReturnValues** : énumération des valeurs

- 1) OpenProject
- 2) OpenProjectErrorMessage

B. OcrTypo

I. Classe **FontModel** : héritage « INotifyPropertyChanged », les informations et éléments de Font Modèle.

Propriétés :

- 1) Répertoire de Font Modèle
- 2) Le nom normalisé de Font Modèle
- 3) Les informations de Font Modèle : Auteur, titre, « place » ?, imprimeur, publication date, publication format, copy Library..., imageFilename, images répertoire, image page, résolution.
- 4) Les Informations du Thumbnail : nom, hauteur, largeur, les positions- X,Y.
- 5) Les informations de typography : type, smallCap, Alphabet, Family, SubFamily, BodyHeight, Thickness
- 6) Description de références et Engraver

II. Classe **IOCR** : interface de OCR Engines intégré dans Retro

Méthodes :

- 1) Récupérer la liste de Font Modèle {*.png, *.xml, *_bw.png}

III. Classe **TemplateMatchingOCREngine** : héritage « IOCR »

Méthodes :

- 1) L'opération de processus OCR : obtenir le nombre de cluster transcrit automatiquement
- 2) Le prétraitement de l'image : binarisation et normalisation
- 3) Comparaison de deux ShapeEoC(bitmap) (même que la fonction dans Clustering.ClusteringTool2)

C. ViewModel

Classe **RetroViewModel** : des méthodes font références aux méthodes de la classe **RetroProject**. Cette classe fait un lien entre IHM et la modèle RetroProjet

3) RetroUtil

A. DynamicSplashScreenNotification :

Définition des notifications pour un event.

B. Image2DisplayTool

Conversion l'image de *System.Drawing.Bitmap* à *System.Windows.Media.Imaging.BitmapSource*