

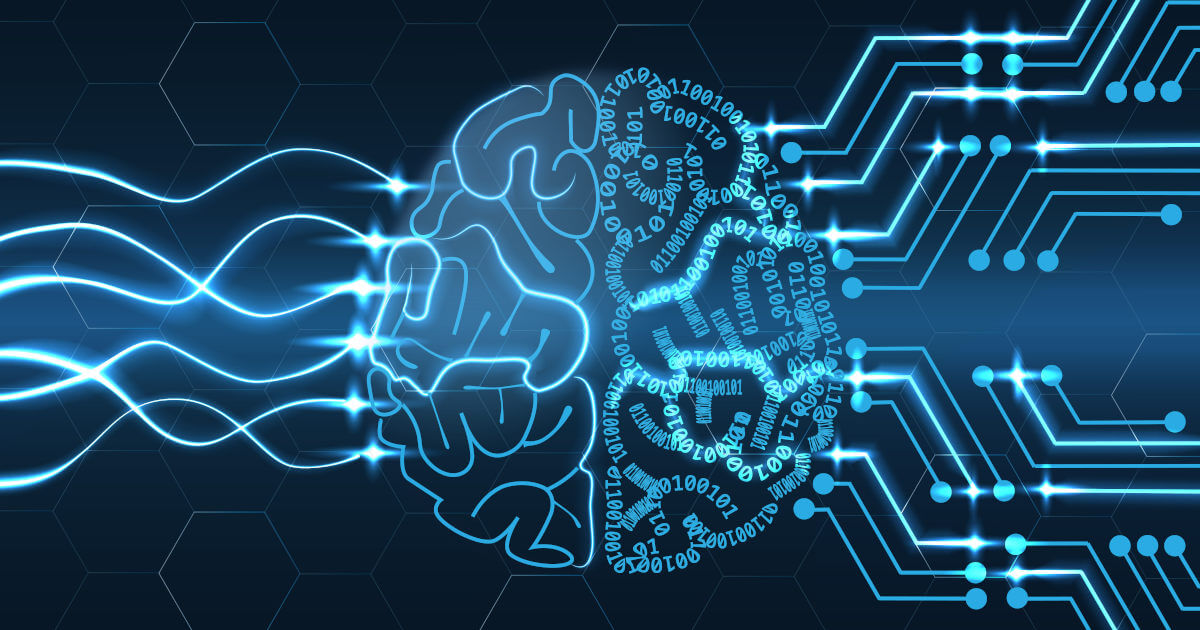
**ROYAUME DU MAROC**

**UNIVERSITE ABDELMALEK ESSAÂDI**

**FACULTE DES SCIENCES ET TECHNIQUES**

**DEPARTEMENT GENIE INFORMATIQUE**

**Projet 1 :**

**Un système d’indexation et de recherche d’images par contenue**

**Réalisé par: Encadré par :**

BENAMAR Zaid AIT KBIR Mohammed

SAMADI Sohaib

FAHAM Hassan

**OBJECTIF**

Notre but dans ce mini projet est de mettre au point une application Web qui permet d’implémenter les fonctionnalités de base d’un système d’indexation et de recherche d’images par le contenu, en utilisant les decripteures de Texture comme les filtres de Gabor et les caractéristiques de Tamura avec retour de pertinence (Re-weighting Type-2).

**INTRODUCTION**

Les technologies peu coûteuses de capture et de stockage d'images ont permis de créer des collections massives d'images numériques. Cependant, à mesure qu'une base de données s'agrandit, la difficulté de trouver des images pertinentes augmente. Pour cette raison, une approche générale de ce problème a été développée, à savoir la recherche d'images basée sur le contenu (CBIR).

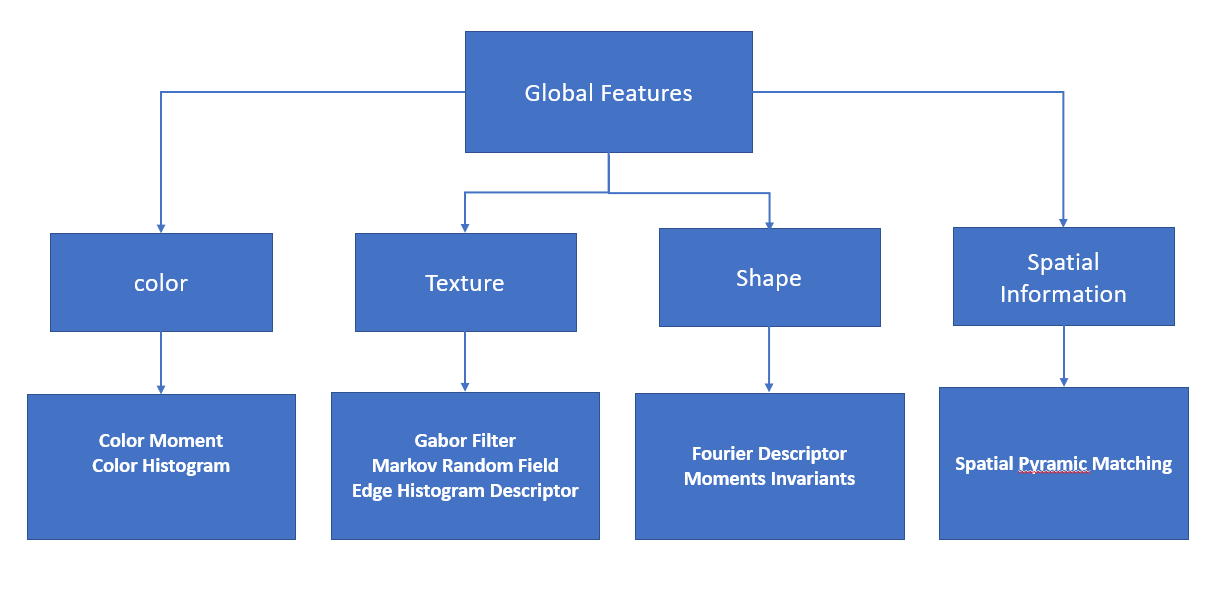
La recherche d'images basée sur le contenu (CBIR) est une méthode de recherche d'images dans une base de données. Dans CBIR, un utilisateur spécifie une image d'interrogation et obtient les images de la base de données similaires à l'image d'interrogation. Pour trouver les images les plus similaires, CBIR compare le contenu de l'image d'entrée aux images de la base de données.

Plus précisément, CBIR compare les caractéristiques visuelles telles que les formes, les couleurs, la texture et les informations spatiales et mesure la similarité entre l'image demandée et les images de la base de données.

Dans ce projet on utilise une base des exemples qui contient 7200 images des objets diffèrent nous avons construire et réaliser une application Web qui permet d’implémenter les fonctionnalités de base d’un système d’indexation et de recherche de ces images par le contenu avec retour de pertinence.

**Méthodes d'extraction de caractéristiques – Texture :**

Les systèmes CBIR doivent effectuer une extraction de caractéristiques, qui joue un rôle important dans la représentation du contenu sémantique d'une image.

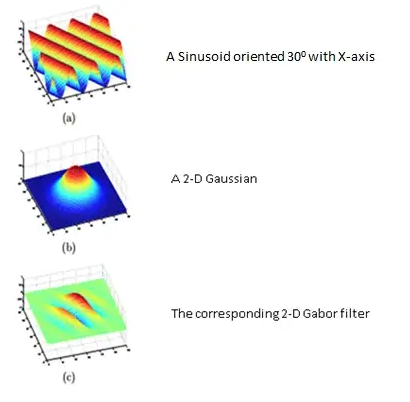


Les textures d'images sont définies comme des images de surfaces naturelles et de polavisuels créés artificiellement. La texture est la régularité de certains motifs formés à partir de l'arrangement des pixels de l'image. Les textures indiquent la nature ou les caractéristiques possédées par une grande zone de sorte que naturellement ces traits peuvent être répétés dans la zone. L'analyse des textures fonctionne en observant le motif des valeurs voisines entre les pixels dans le domaine spatial. En outre, les textures décrivent généralement les caractéristiques intrinsèques de l'image en mesurant la rugosité, la granularité et la régularité de la structure disposition structurelle des pixels. L'aspect textural d'une image peut être utilisé comme base pour la segmentation, la classification ou l'interprétation de l'image. Les descripteurs de texture fournissent des mesures de propriétés telles que la douceur, la grossièreté et la régularité.

Les approches statistiques permettent de caractériser la texture en tant que lisse, grossière, granuleuse et ainsi de suite. La détermination de la texture est convient parfaitement à la recherche d'images médicales.

**Filtres de Gabor :**

Le filtre de Gabor, du nom de Dennis Gabor, est un filtre linéaire utilisé dans une myriade d'applications de traitement d'images pour la détection des bords, l'analyse de la texture, l'extraction de caractéristiques, etc. Les caractéristiques de certaines cellules du cortex visuel de certains mammifères peuvent être approximées par ces filtres. Il a été démontré que ces filtres possèdent des propriétés de localisation optimales dans les domaines spatial et fréquentiel et sont donc bien adaptés aux problèmes de segmentation de texture. Les filtres de Gabor sont des classes spéciales de filtres passe-bande, c'est-à-dire qu'ils autorisent une certaine "bande" de fréquences et rejettent les autres. Un filtre de Gabor peut être considéré comme un signal sinusoïdal de fréquence et d'orientation particulières, modulé par une onde gaussienne. Un tel filtre de Gabor 2D est illustré à la figure 1.

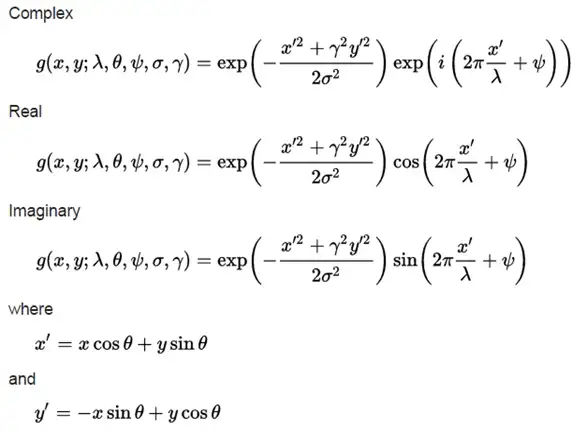


Pour commencer, les filtres de Gabor sont appliqués aux images à peu près de la même manière que les filtres classiques. Nous disposons d'un masque qui représente le filtre. Par masque, nous voulons dire que nous disposons d'un tableau (généralement un tableau 2D puisqu'il s'agit d'images 2D) de pixels dans lequel chaque pixel se voit attribuer une valeur (appelée "poids"). Ce tableau est glissé sur chaque pixel de l'image et une opération de convolution est effectuée.

Lorsqu'un filtre de Gabor est appliqué à une image, il donne la réponse la plus élevée aux bords et aux points où la texture change. Les images suivantes montrent une image de test et sa transformation après l'application du filtre.



Certains paramètres déterminent la manière dont le filtre de Gabor sera utilisé et les caractéristiques auxquelles il répondra. Un filtre de Gabor 2D peut être considéré comme un signal sinusoïdal de fréquence et d'orientation particulières, modulé par une onde gaussienne. Le filtre possède une composante réelle et une composante imaginaire représentant des directions orthogonales. Les deux composantes peuvent être formées en un nombre complexe ou utilisées individuellement. Les équations sont présentées ci-dessous :



Dans l'équation ci-dessus,

**λ** - La longueur d'onde de la composante sinusoïdale.

**Ө** - L'orientation de la normale aux bandes parallèles de la fonction de Gabor.

**Ψ** - Le décalage de phase de la fonction sinusoïdale.

**σ** - Le sigma/écart-type de l'enveloppe gaussienne.

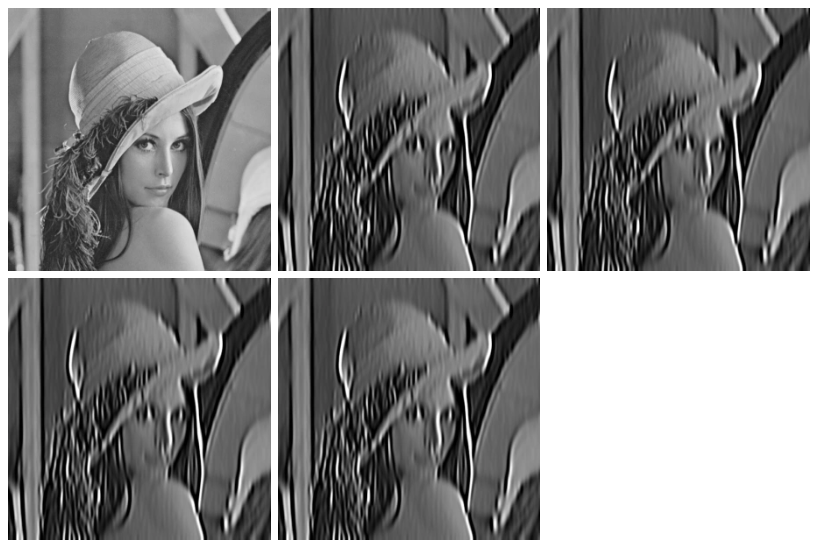
**ɣ** - Le rapport d'aspect spatial et spécifie l'ellipticité du support de la fonction de Gabor.

Lorsqu'on dit qu'un filtre répond à une caractéristique particulière, cela signifie que le filtre a une valeur distinctive à l'emplacement spatial de cette caractéristique. Ces paramètres affectent la sortie d'un filtre de Gabor. Dans OpenCV Python, voici la structure de la fonction qui est utilisée pour créer un noyau de Gabor.

cv2.getGaborKernel(ksize, sigma, theta, lambda, gamma, psi, ktype)

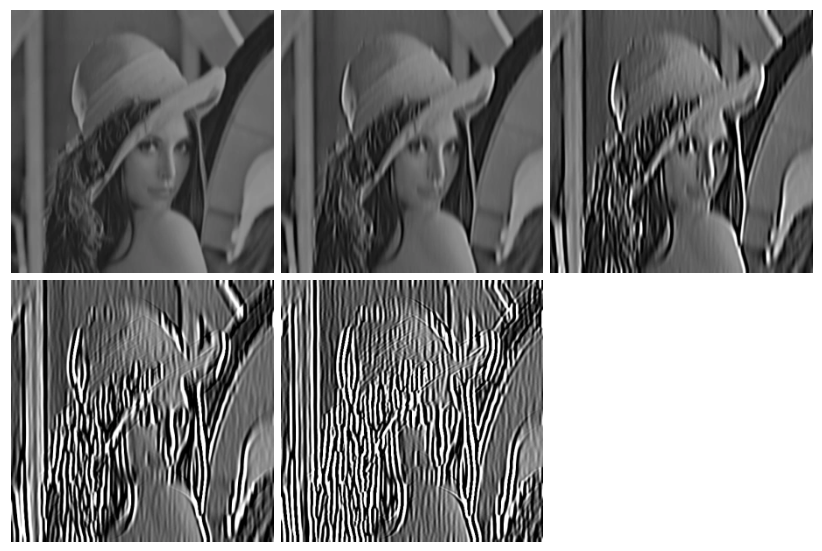
**ksize** est la taille du noyau de Gabor. Si ksize = (a, b), nous avons alors un noyau de Gabor de taille a x b pixels. Comme pour de nombreux autres noyaux de convolution, ksize est de préférence impair et le noyau est carré (par souci d'uniformité).

En faisant varier ksize, la taille du noyau de convolution varie. Nous observons qu'il n'y a aucun effet de la taille du noyau de convolution sur l'image de sortie. Cela implique également que le noyau de convolution est invariant à l'échelle, puisque la mise à l'échelle de la taille du noyau est analogue à la mise à l'échelle de la taille de l'image. Voici quelques résultats avec une taille de noyau variable.

****

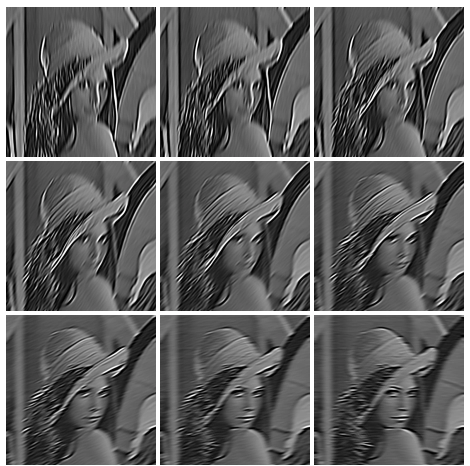
**sigma** est l'écart-type de la fonction gaussienne utilisée dans le filtre de Gabor.

Ce paramètre contrôle la largeur de l'enveloppe gaussienne utilisée dans le noyau de Gabor. Voici quelques résultats obtenus en faisant varier ce paramètre.

****

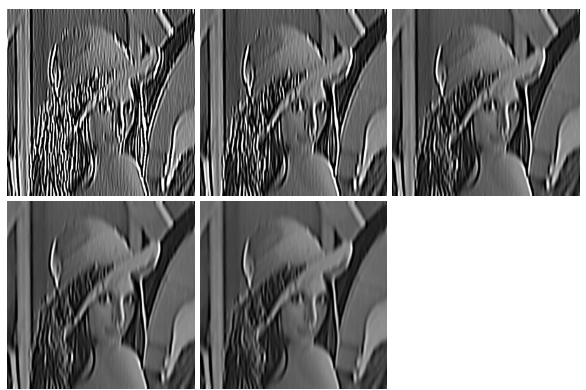
**thêta** est l'orientation de la normale aux bandes parallèles de la fonction de Gabor.

Il s'agit peut-être de l'un des paramètres les plus importants du filtre de Gabor. Ce paramètre détermine le type de caractéristiques auxquelles le filtre répond. Par exemple, donner à thêta une valeur de zéro signifie que le filtre ne réagit qu'aux caractéristiques horizontales. Ainsi, afin d'obtenir des caractéristiques à différents angles dans une image, nous divisons l'intervalle entre 0 et 180 en 16 parties égales, et calculons un noyau de Gabor pour chaque valeur de thêta ainsi obtenue. Notez que nous avons choisi 16 simplement parce que c'était la valeur par défaut dans l'implémentation OpenCV. Ces valeurs de paramètres peuvent être modifiées pour répondre à des besoins spécifiques. Voici les résultats de la variation de thêta sur l'image d'entrée ci-dessus.

****

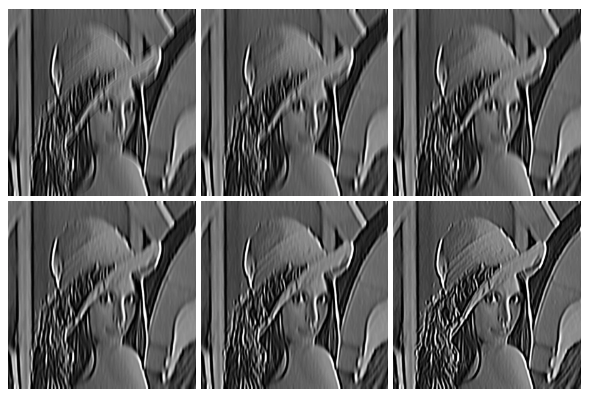
**lambda** est la longueur d'onde du facteur sinusoïdal dans l'équation ci-dessus.

Voici la variation avec lambda (thêta est mis à zéro).

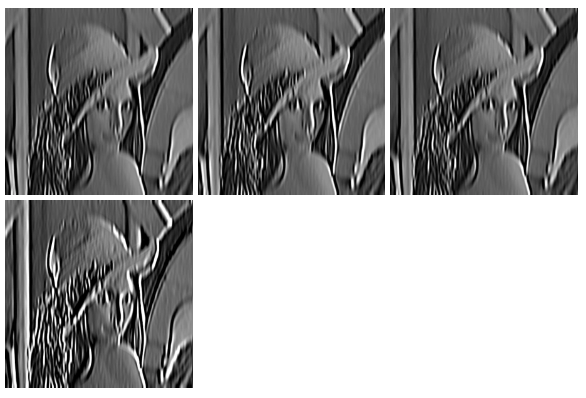


**gamma** est le rapport d'aspect spatial.

Le gamma contrôle l'ellipticité de la gaussienne. Lorsque gamma = 1, l'enveloppe de la gaussienne est circulaire.



**psi** est le décalage de phase.

****

**ktype** indique le type et la plage de valeurs que chaque pixel du noyau de Gabor peut contenir.

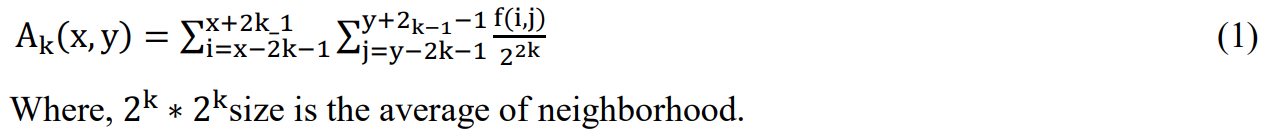
**Filtres de Tamura :**

la méthodologie de Tamura approche l’extraction et l’analyse d’une texture en concevant des caractéristiques qui correspondent à la perception visuelle humaine.

Selon l'analyse quantitative, l'une des premières descriptions données par le Tamura a proposé six propriétés texturales et a donné des descriptions communes à tous les modèles de texture dans les images photographiques de Broadtz. Il s'agit de six caractéristiques de texture différentes données par Tamura : grossièreté, contraste, directionnalité, ressemblance des lignes, régularité et rugosité.

**Grossièreté**e :

La grossièreté est essentiellement liée à la distance en niveaux de gris des variations spatiales, qui est implicitement liée à la taille des éléments primitifs formant la texture. Elle est en relation directe avec l'échelle et les taux de répétition et constitue la caractéristique la plus fondamentale de la texture. Une image contient des motifs de textures répétés à différentes échelles, la grosseur vise à identifier la plus grande taille à laquelle une texture existe, même si une micro texture plus petite existe.



****

Cette équation (2) calcule la différence entre les paires de moyennes correspondant à des voisinages qui ne se chevauchent pas.

**Contraste** :

Le contraste mesure la distribution des niveaux de gris qui varie dans une image et dans quelle mesure sa distribution est biaisée vers le noir ou le blanc. Les moments centraux de second ordre et de quatrième ordre normalisé des niveaux de gris sont utilisés pour définir le contraste.



où, 𝜇4 est le quatrième moment autour de la moyenne et 2 est la variance. n=1/4 pour donner la valeur la plus proche selon Tamura.

**Directionnalité** :

La directionnalité d'une image est mesurée par la distribution de fréquence des bords locaux orientés par rapport à leurs angles directionnels. Il s'agit d'une propriété globale sur une région. Cette caractéristique de texture donnée par Tamura ne fait pas la différence entre les orientations ou les motifs mais mesure le degré total de directionnalité dans une image, ce qui est donné par Directionnalité. C'est la caractéristique la plus importante donnée par Tamura sur la matrice pour distinguer d'une autre image le degré d'uniformité de la région.



où, np, est le nombre de sommets, ap, est la position du sommet, wp, est la plage des angles attribués au Pième sommet, r désigne un facteur de normalisation lié aux niveaux de quantification des angles a, et a désigne l'angle directionnel quantifié, 𝐻𝐷𝑖𝑟𝑒𝑐𝑡𝑖𝑜𝑛𝑎𝑙𝑖𝑡𝑦, est l'histogramme des valeurs de direction quantifiées ,a est construit en comptant le nombre de pixels du bord avec les angles directionnels correspondants.

**Ressemblance des lignes :**

La ressemblance des lignes dans une image est la coïncidence moyenne de la direction des bords qui coïncident dans les paires de pixels séparés par une distance le long de la direction du bord dans chaque pixel.

**Régularité :**

La régularité mesure la régularité d'un motif ou d'un élément similaire présent dans une image.



**Rugosité :**

La rugosité est la somme des mesures de contraste et de rugosité.

Rugosité= Contraste + Grossièreté

Dans la plupart des cas, pour un système CBIR, seules les trois premières caractéristiques sont utilisées car elles capturent les attributs perceptifs de haut niveau d'une texture et sont également utiles pour la navigation dans les images.

**Retour de pertinence – Reweighting type 2 :**

L'objectif de la mise à jour des poids est de mettre l'accent sur les paramètres les plus discriminants. En pratique, l'idée est d'effectuer une sélection dynamique des caractéristiques, en fonction des commentaires des utilisateurs. L'algorithme de repondération des caractéristiques utilisé dans ce travail est similaire à celui proposé dans Mattia et Francesco (2010) et Wu et Zhang (2002) et est basé sur un ensemble de caractéristiques statistiques.

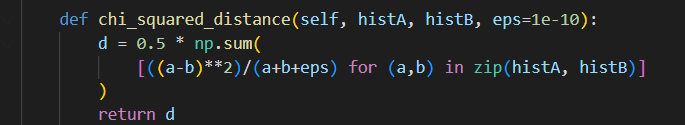
Wu et Zhang ont utilisé des images pertinentes et non pertinentes pour mettre à jour les pondérations. Ils ont utilisé un ratio discriminant pour déterminer la capacité d'une composante caractéristique à séparer les images pertinentes de celles qui ne le sont pas.

**Métrique de similarité**

Maintenant que nous avons des volumes massifs de vecteurs de caractéristiques. Mais comment allez-vous les comparer ? Pour cette raison il faut définir une mesure de similarité qui quantifie la similarité entre une image de la base de données et l'image demandée. D’autre part la sélection de la bonne mesure de similarité a toujours été une tâche difficile car il dépend fortement (1) de l’ensemble de données et (2) des types de caractéristiques extraites.

Dans ce projet nous extrayons les vecteurs associés à l'image indexée, puis nous les comparons aux caractéristiques de l'image de la requête à l'aide de **la distance chi carrée.**

* **Implémentation python :**

****

**Recherche des images par contenue**

L'étape finale consiste à effectuer une recherche réelle. Un utilisateur soumettra une image de requête à notre système (à partir d'un formulaire de téléchargement ou via une application mobile) et après il faut (1) extraire les caractéristiques de cette image de requête et ensuite (2) appliquer la fonction de similarité pour comparer les caractéristiques de la requête aux caractéristiques déjà indexées.

La recherche est effectuée en utilisant une base de données d'images présentant les caractéristiques suivantes :

a) Les données d'images observées s'élèvent à 7200 pièces.

b) Chaque image est au format png

c) La résolution des images recherchées et interrogées est de 128x128 pixels

il y a plusieurs processus qui seront effectués à partir du test de l'image interrogée et de l'analyse de la texture jusqu'à ce que les résultats correspondent au test de l'image interrogée. Les processus sont :

a) Saisir l'image d'input.

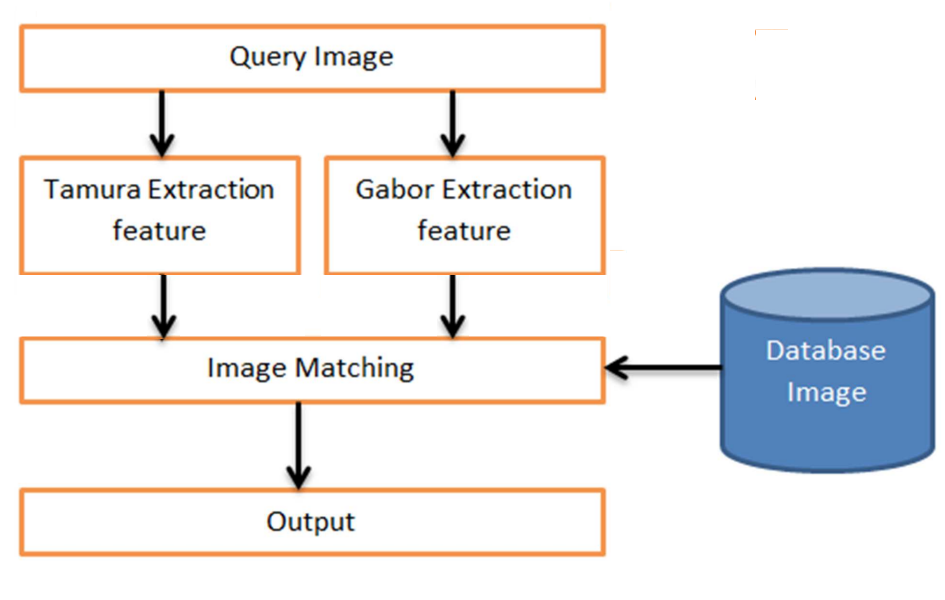
b) Extraction de la texture de l'image input en utilisant les caractéristiques de Tamura.

c) Extraction de la texture de l'image input en utilisant les filtres de Gabor.

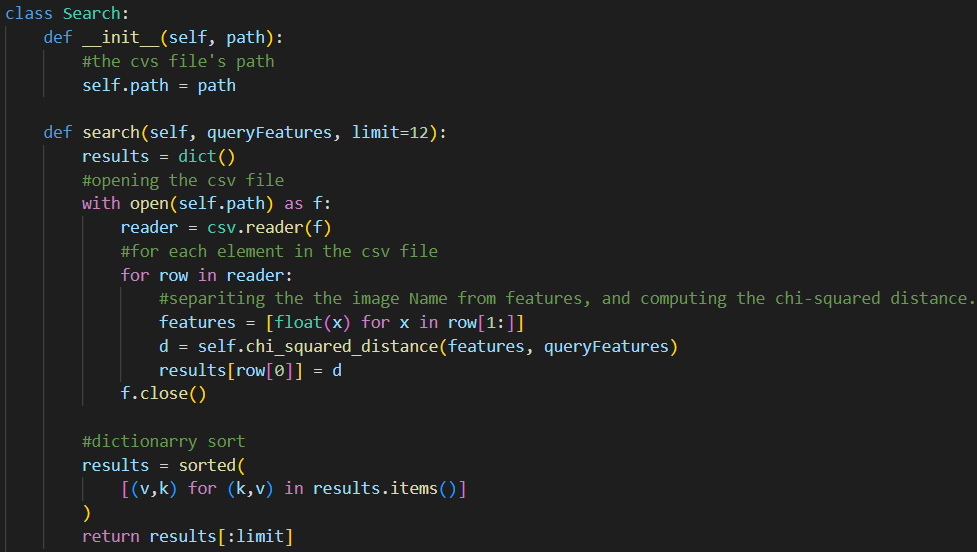
d) Faire correspondre l'image à la base de données

e) Afficher les résultats de la comparaison.

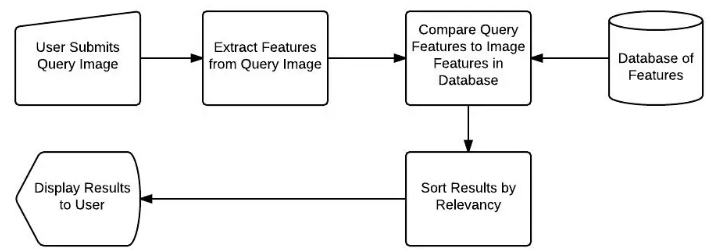
Voici une image ou un modèle d'organigramme du processus qui sera effectué :



* **Implémentation python :**



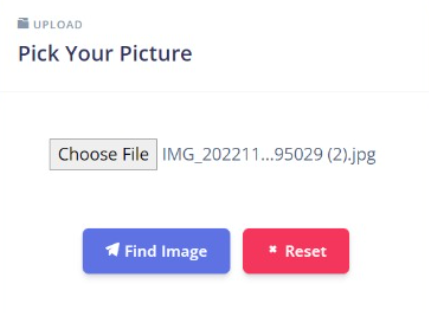
**Architecture de moteur de recherche**



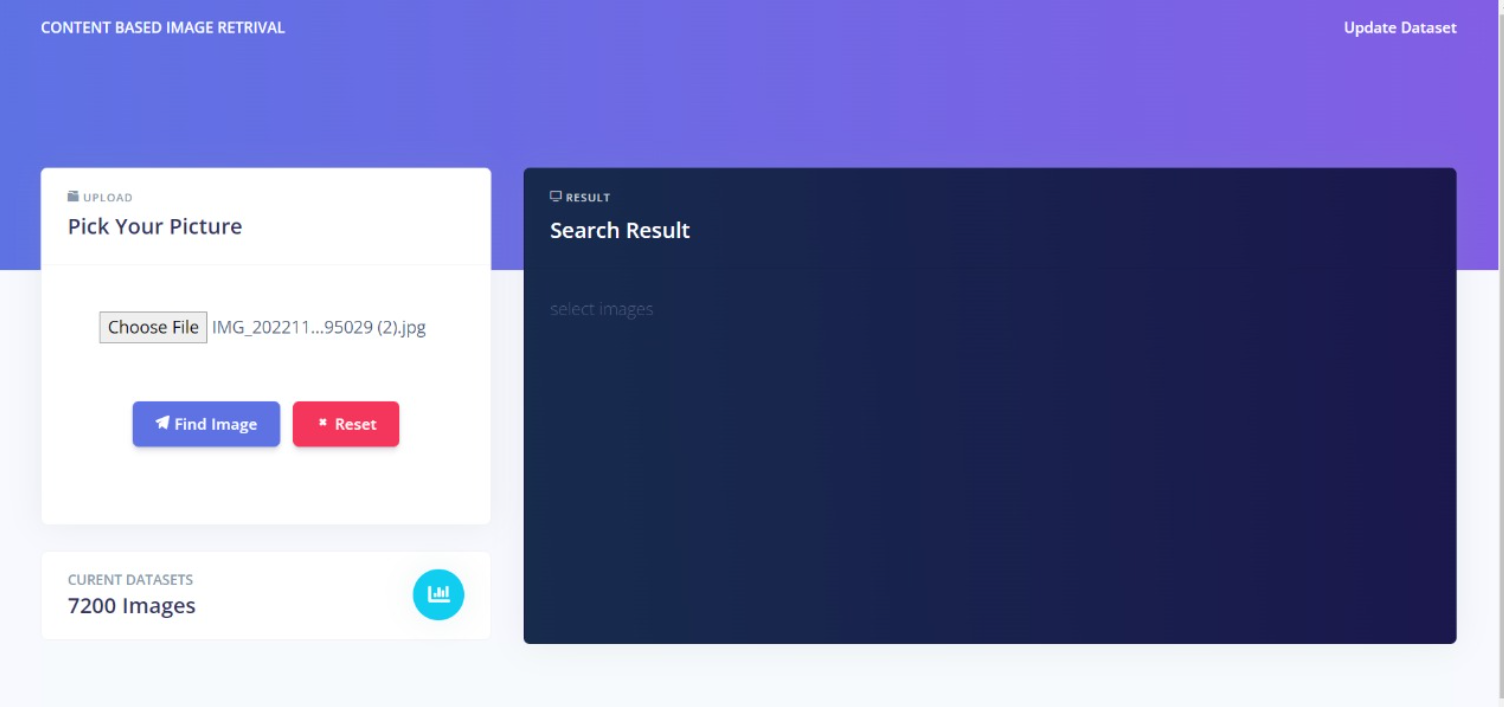
Tout d'abord, un utilisateur doit soumettre une image de requête à notre moteur de recherche d'images. Nous prenons ensuite cette image et en extrayons les caractéristiques. Ces "caractéristiques de la requête" sont ensuite comparées aux caractéristiques des images que nous avons déjà indexées dans notre jeu de données. Enfin, les résultats sont triés par pertinence et présentés à l'utilisateur.

**Réalisation d’une application Web basée sur Framework Flask**

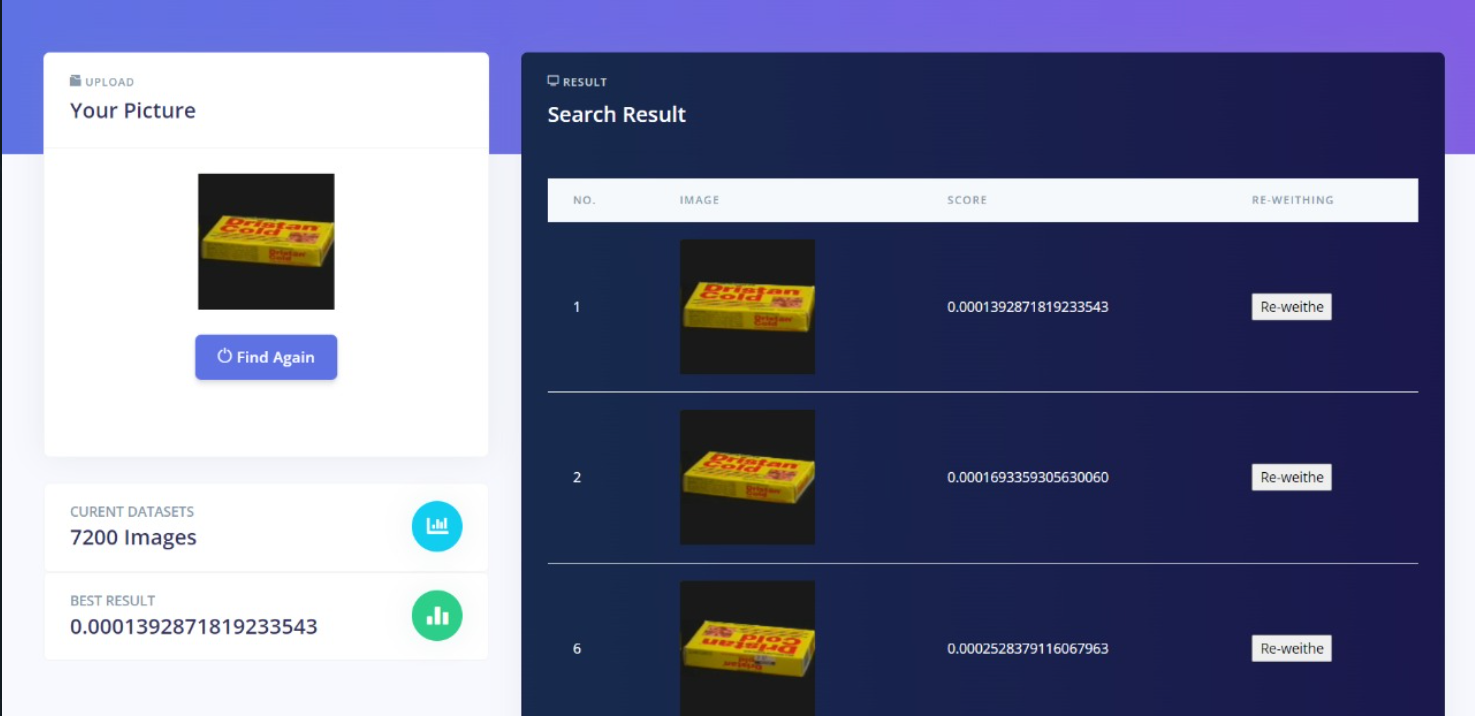
A partir d'un formulaire de téléchargement l’utilisateur doit choisir une image.



Ensuite le moteur de recherche va extraire les caractéristiques de l’image choisie par l’utilisateur et les comparées aux caractéristiques des images que nous avons déjà indexées dans le fichier index.csv.



* **Résultat de recherche :**

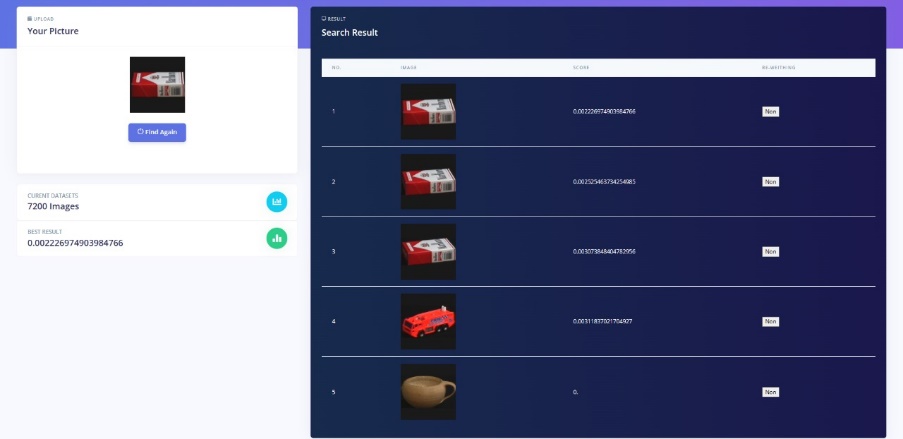


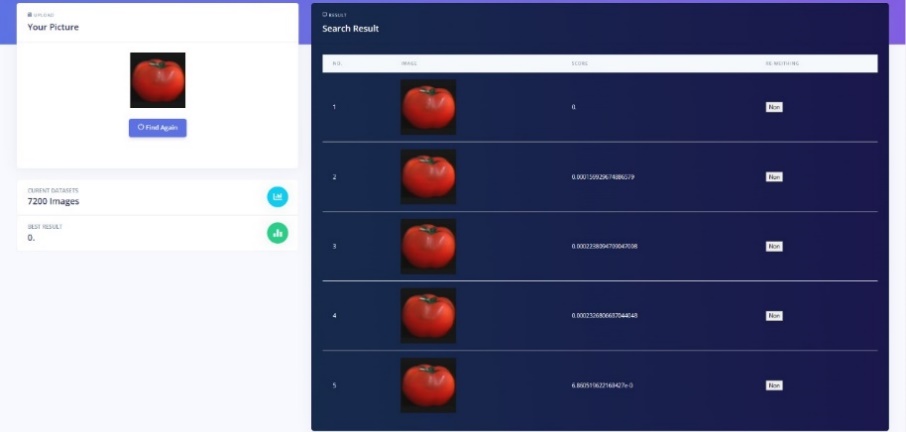
On peut même cliquer sur « **Re-weighte** » bouton s’il s’agit d’un résultat faux pour le retour de pertinence et l’ajustement des weights afin d’avoir des résultat correct et similaire.

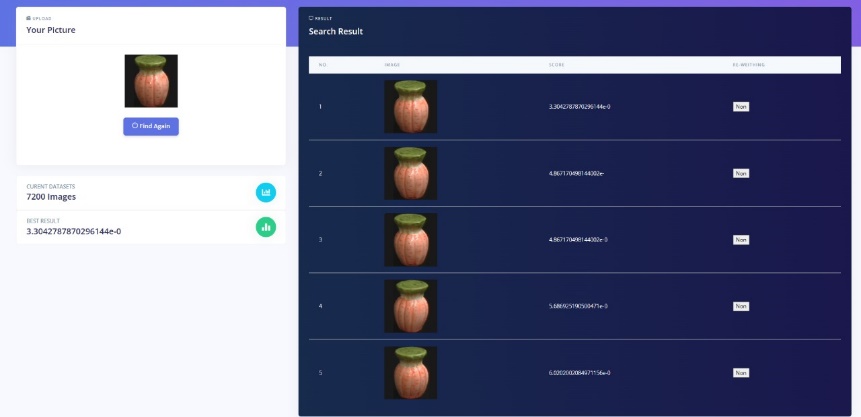
**Evaluation des performances du système réalisé**

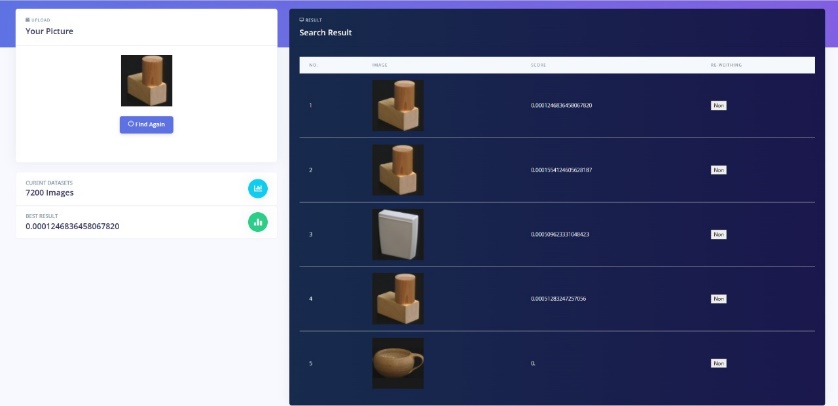
Pour évaluer les performances d'un système d’indexation et de recherche par le contenu de dans une base d’exemples de modèles 3D et utilisant des fichiers .obj, vous pouvez utiliser les mesures suivantes :

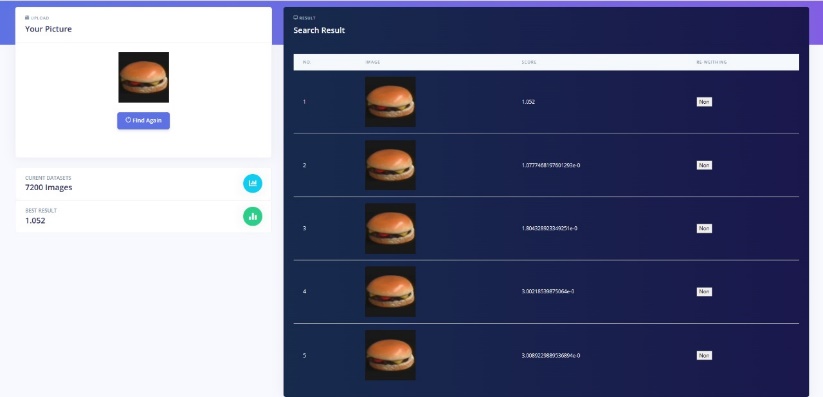
1. **Précision** : Cette métrique mesure la proportion de modèles récupérés qui sont pertinents pour le modèle de la requête.
2. **Rappel** : Cette métrique mesure la proportion de modèles pertinents qui sont récupérés par le système.
3. **Score F1 :** Il s'agit de la moyenne harmonique de la précision et du rappel, et constitue une bonne mesure globale des performances du système.
4. **Précision moyenne (MAP) :** Cette métrique mesure la précision moyenne du système sur l'ensemble des requêtes.











Precision = (Number of relevant models retrieved) / (Number of models retrieved)

Recall = (Number of relevant models retrieved) / (Number of relevant models in the dataset)

F1 Score = 2 \* (Precision \* Recall) / (Precision + Recall)

La précision moyenne (MAP) est une mesure des performances d'un système qui tient compte à la fois de la pertinence des modèles récupérés et de leur classement pour un ensemble de requêtes. Elle est couramment utilisée dans les systèmes de recherche d'informations et de recommandation. La formule de la MAP est la moyenne de la précision à chaque point de récupération pertinent pour un ensemble donné de requêtes.

Pour calculer la MAP à l'aide de l'exemple que vous avez fourni, vous devez d'abord calculer la précision moyenne (AP) pour chaque requête.

La formule de la précision moyenne (AP) est la somme de la précision à chaque point de récupération pertinent, divisée par le nombre de modèles pertinents pour la requête. AP = (1/|Modèles pertinents pour la requête|) \* (P(1) + P(2) + .... + P(n))

où P(n) est la précision au nième point de récupération pertinent, et |Modèles pertinents pour la requête| est le nombre total de modèles pertinents pour la requête.

Pour la requête 1, 5 modèles ont été récupérés avec le classement suivant (pertinent, pertinent, pertinent, non pertinent, non pertinent), le PA serait de (1+1+1) /3 = 1

Pour la requête 2, puisque les 5 modèles récupérés sont pertinents, le PA serait de 1,0.

Pour les requêtes 3 et 5, le PA serait de 1,0, puisque les 5 modèles récupérés sont pertinents.

Pour la requête 4, 5 modèles ont été récupérés avec le classement suivant (pertinent, pertinent, non pertinent, pertinent, non pertinent), le calcul serait AP = (1/4) \* (P(1) + P(2) + P(3) + P(4)).

Pour calculer P(n) pour la requête 4 : P(1) = 1, P(2) = 2/2, P(3) = 2/3, P(4) = 3/4 AP = (1/4) \* (1 + 2/2 + 2/3 + 3/4) = 0,8333

Enfin, vous devez calculer la moyenne de tous les PA pour chaque requête afin d'obtenir la MAP : MAP = (AP(requête 1) + AP(requête 2) + AP(requête 3) + AP(requête 4) + AP(requête 5)) / 5 = (1 + 1 + 1 + 0.8333 + 1) / 5 = 0.9666

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | Query 1 | Query 2 | Query 3 | Query 4 | Query 5 |
| Precision | 3/5=0.6 | 5/5=1 | 5/5=1 | 3/5=0.6 | 5/5=1 |
| Recall | 3/72=0.04 | 5/72=0.07 | 5/72=0.07 | 3/72=0.04 | 5/72=0.07 |
| F1 Score | 0.075 | 0.13 | 0.13 | 0.075 | 0.13 |
| Average precision | 1 | 1 | 1 | 0.833 | 1 |
| MAP |  |  | 0.966 |  |  |

**CONCLUSION**

Ce travail a pour objectif de concevoir et faire face aux différentes techniques d’indexation et de recherche d’images par le contenu avec retour de pertinence, qui permet trouver des images similaires à l'aide Méthodes d'extraction de caractéristiques.

Pour pouvoir compléter notre projet, Nous avons détaillé les différentes étapes implémentation, de conception et réalisation de l’application web.

Durant ces parties, j’avais présenté la conception détaillée, à la fois, de notre projet, d’autre part j’avais aussi présenté les tâches qui a été accompli pour chaque partie, ensuite, j’ai présenté quelques captures d’écran montrant le bon fonctionnement du moteur de recherche.

**REFERENCES**

<https://medium.com/@anuj_shah/through-the-eyes-of-gabor-filter-17d1fdb3ac97>

https://cvtuts.wordpress.com/tag/gabor-wavelet/

<https://www.freedomvc.com/index.php/2021/10/16/gabor-filter-in-edge-detection/>

Wang X., Ding X., Liu C, Gabor filters-based feature extraction for character recognition, Pattern Recognition, 38 (3) (2005), Pages 369-379

W.K. Kong, D. Zhang, W. Li, Palmprint feature extraction using 2-D Gabor filters, Pattern Recogn., 36 (10) (2003), pp. 2339-2347

Bahtiar Imran. Content-Based Image Retrieval Based on Texture and Color Combinations Using Tamura Texture Features and Gabor Texture Methods. American Journal of Neural Networks and Applications. Vol. 5, No. 1, 2019, pp. 23-27. doi: 10.11648/j.ajnna.20190501.14

A. Halim, et al., “Aplikasi Image Retrieval Menggunakan Kombinasi Metode Color Moment dan Gabor Texture, “JSM STMIK Mikroskil, vol. 14, October 2013, pp. 109-117.

Thawari. P. B. and Janwe. N. J., “CBIR BASED ON COLOR AND TEXTURE”, International Journal of Information Technology and Knowledge Management, Vol. 4, No. 1, pp. 129-132.

S. Mori Tamura and T. Yamawaki, “Textural Features Corresponding to Visual Perception”, IEEE Transactions on Systems, Man and Cybernetics, vol. smc-8, no. 6, (1978) June, pp. 460-473

Neelima. B. and Punit. K. J., “A Comparative Study on Feature Extraction using Texture and Shape for Content Based Image Retrieval”, Madhav Institute of Technology and Science, Vol.80 (2015), pp.41-52