

TD AVIO





Classification d'images

BOUSSOT Valentin, HEMON Cédric

26 mars 2020

Objectifs

L'objectif principal est de classifier un ensemble d'images dans 4 catégories sachant que cet ensemble diffère d'une image de base représentant sa catégorie par des transformations géométriques RTS (Rotation, Changement d'échelle, Translation), des techniques d'apprentissage automatique supervisé sont utilisées pour ce problème de classification. De manière plus précise l'objectif est double, le premier est d'entraîner un modèle le plus efficient possible, c'est-à-dire une prédiction rapide et précise. Le deuxième est d'être capable de donner une estimation de la performance du modèle la plus fiable possible.

Image de base				
Labels	Club	Diamonds	Heart	Spade

L'ensemble des images sont binarisées et de même taille.

Génération de l'ensemble d'images d'entraînement

Les images sont générées par une combinaison de ces transformations pour un total de 1728 images

Labels	Club	Diamonds	Heart	Spade	Commentaire
Rotation	8	8	8	8	$\delta\theta = \frac{\pi}{4}$
Zoom	6	6	6	6	Plage $[0.5, 3]$ $\delta z = 0.5$
Translation	9	9	9	9	8 points cardinaux + centre

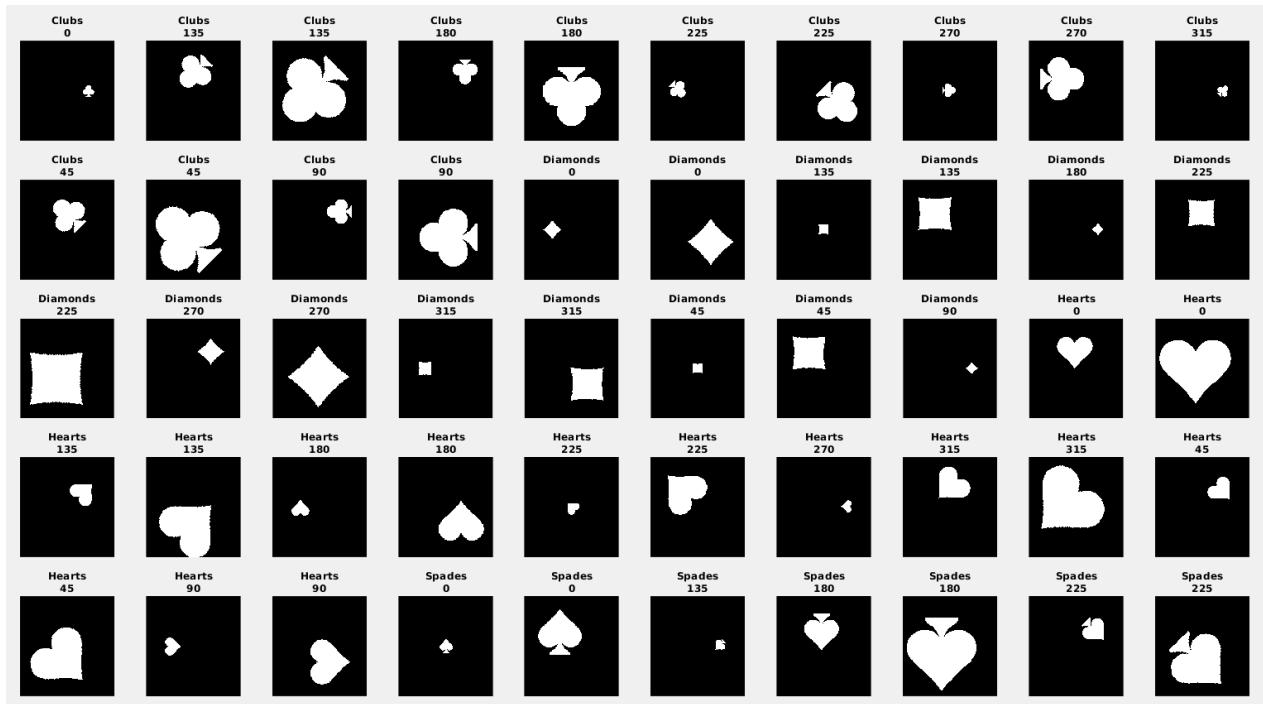


FIGURE 1 – Exemple d’images de l’ensemble généré

Listing 1 – Fonction de transformation géométrique d’une image

```

1 function [result, dim] = ImTransform(image, R)
2     [m,n] = size(image);
3     [X,Y] = meshgrid(1:m,1:n);
4     % Matrice de position des pixels
5     param = [X(:) Y(:) ones(size(Y(:),1), 1)];
6     % Application de la matrice de transformation
7     source = uint8((param-[m/2 n/2 0])*R'+[m/2 n/2 0]);
8     % Les index qui dépasse les dimensions son mis à 1
9     source(~(all(source(:,1:2) >= 1, 2) & all(source(:,1:2) <= [m n], 2)), :) = 1;
10    % Génération de la nouvelle image avec les indice modifié
11    result(sub2ind(size(image), X(:), Y(:))) = ...
12    image(sub2ind(size(image), source(:,1), source(:,2)));
13    result = reshape(result, m, n);
14    % Dimension de l'objet pour les positions
15    [~, col1] = find(result);
16    [~, col2] = find(result');
17    dim = [min(col1) min(col2) max(col1) max(col2) n-max(col1) m-max(col2)];
18 end

```

Les formes étant simple nous avons implémenté un algorithme plus rapide qui ne nécessite pas d’interpolation.

Génération des caractéristiques

Les descripteurs suffisamment invariant RTS et qui discrimine suffisamment les images de base sont apriorie les mieux adaptés pour résoudre ce problème de classification. Mais on nous propose de résoudre ce problème avec un espace de représentation généré uniquement par ces descripteurs ; $m_{p,q}, \mu_{p,q}, \eta_{p,q}$ respectivement les moments cartésiens, centraux (**Invariant : translation**) et centraux normalisés (**Invariant : translation, changement d'échelle**) avec $0 \leq p \leq P$ et $0 \leq q \leq Q$.

Listing 2 – Fonction de génération de l'espace de représentation

```
1 function [features, featuresNorm] = GenFeatures(data, param)
2     n = size(data,3);
3     for i = 1:n
4         e = 1;
5         for p = param'
6             % Calcule des moments d'ordre p,q + type
7             features(i, e) = Moment(data(:,:,i), p);
8             e = e + 1;
9         end
10        disp(['Progression : ', sprintf('%.2f', i/n*100) , '%']);
11    end
12    % Z-Score
13    featuresNorm = Normalise(features); % Normalisation des features
14    featuresNorm = reshape(featuresNorm(~isnan(featuresNorm(:)))...
15    , size(featuresNorm,1), []);
16 end
```

Les caractéristiques n'ayant pas la même échelle de valeurs il est bon pour notre étude de les standardiser (z-score)

$$Y^i = \frac{X^i - \mu_{X^i}}{\sigma_{X^i}} \quad \text{X la } i^{eme} \text{ caractéristiques de l'ensemble des features vu comme une variable aléatoire}$$

Prendre soin que $\sigma_{X^i} \neq 0$ cas de $\eta_{0,0}$

Nous avons décrit les images avec les paramètres P et Q = 10, chaque image sera donc décrite par 362 moments ce qui génère un espace de représentation initiale $E_{1728 \times 362}$

Problématique et réduction de la dimensionnalité

La principale difficulté est la grande dimension de l'espace de représentation, en effet avec le nombre important de caractéristiques on prend le risque que le modèle soit en sur-apprentissage, il sera donc difficile d'évaluer la performance réelle du modèle du fait qu'il n'y a pas de garantie sur sa capacité à généraliser ce qui impacte les deux objectifs. De plus sans une étude au préalable de ses caractéristiques il sera difficile de stratifier correctement les images de tests et d'entraînements pour l'estimation du pouvoir de prédiction du model.

Cette identification des difficultés émerge d'expérimentation effectuée en TD. Dans lequel un algorithme de sélection et un classifieur K-NN a été développé. La performance du modèle était estimée avec une validation croisée à 3 folds. La performance moyenne était au-dessus de 60 % mais avec une variance de performance entre folds élevée.

La première étape sera donc de réduire la dimensionnalité de l'espace de représentation en essayant de conserver les caractéristiques très pertinentes tous en écartant les caractéristiques peu informatives.

Il existe de nombreuses méthodes pour trouver ce sous-ensemble de caractéristique pertinente. Clairement une recherche exhaustive de tous les sous-ensembles de caractéristiques va nous garantir de trouver un sous-ensemble optimal sur un classifieur donné. Dans notre cas cette approche pose problème, en effet on essaye d'explorer une large partie des moments, les combinaisons de cette caractéristique ainsi créée croît exponentiellement ce qui a un coût en temps non négligeable, de plus la question de savoir comment stratifier les images de tests et d'entraînement se pose toujours.

Pour répondre à ces difficultés on propose cette méthodologie :

1) Implémenter un algorithme de sélection sur une approche Filter afin de sélectionner les caractéristiques les plus prometteuses. Ce qui nous donnera les premières invariance et régularité au sein de l'ensemble d'entraînement. Cette action va permettre de réduire une première fois le domaine de recherche des caractéristiques sur un ensemble d'images d'entraînement plus réduit. Elle va permettre de garantir une bonne généralisation du modèle et nous fournir une idée sur la stratification à adopter.

2) Sélectionner un sous-ensemble optimal de caractéristique avec une approche Wrapper sur un classifieur K-NN.

3) Vérifier la bonne généralisation et améliorer l'efficacité du modèle.

Algorithme de sélection

On va utiliser tous d'abord un algorithme de sélection par une approche filtre qui se base sur des critères statistique de l'espace de représentation, afin de construire une mesure de pertinence pour sélectionner un sous-ensemble de caractéristique, indépendamment du classifieur qui le suit. Ce choix d'approche est pertinent dans la mesure où on essaye d'explorer une large partie des moments.

Nous avons choisi le critère de Fisher qui permet de mesurer le degré de séparabilité des classes qui est le rapport entre la variance interclasse et la variance intra-classe.

$$F_{X^i} = \frac{\sum_{c=1}^4 n_c \cdot (\mu_{X_c^i} - \mu_{X^i})^2}{\sum_{c=1}^4 n_c \cdot \sigma_{X_c^i}^2}$$

avec n_c , $\mu_{X_c^i}$ et $\sigma_{X_c^i}$ représentent respectivement l'effectif, la moyenne et l'écart type de la i^{eme} caractéristique au sein de la classe c . μ_{X^i} est la moyenne globale de la i^{eme} caractéristique.

Les classes étant identiquement distribuées au sein de l'ensemble d'entraînement l'effectif, n_c peut être simplifié. Le score est évalué après normalisation $\Rightarrow \mu_{Y^i} = 0$

$$F_{Y^i} = \frac{\sum_{c=1}^4 \mu_{Y_c^i}^2}{\sum_{c=1}^4 \sigma_{Y_c^i}^2}$$

L'algorithme de sélection n'est pas affecté par la normalisation en amont pour s'en convaincre :

$$\begin{aligned} \mu_{Y_c^i}^2 &= \mathbb{E}[Y_c^i]^2 = \mathbb{E}\left[\frac{X_c^i - \mu_{X^i}}{\sigma_{X^i}}\right]^2 = \frac{(\mathbb{E}[X_c^i] - \mu_{X^i})^2}{\sigma_{X^i}^2} = \frac{(\mu_{X_c^i} - \mu_{X^i})^2}{\sigma_{X^i}^2} \\ \sigma_{Y_c^i}^2 &= \mathbb{E}[(Y_c^i)^2] - \mathbb{E}[Y_c^i]^2 = \mathbb{E}\left[\left(\frac{X_c^i - \mu_{X^i}}{\sigma_{X^i}}\right)^2\right] - \mathbb{E}\left[\frac{X_c^i - \mu_{X^i}}{\sigma_{X^i}}\right]^2 = \\ &= \frac{\mathbb{E}[(X_c^i)^2] - 2\mu_{X^i}\mathbb{E}[X_c^i] + \mu_{X^i}^2}{\sigma_{X^i}^2} - \frac{\mathbb{E}[X_c^i]^2 - 2\mu_{X^i}\mathbb{E}[X_c^i] + \mu_{X^i}^2}{\sigma_{X^i}^2} = \frac{\mathbb{E}[(X_c^i)^2] - \mathbb{E}[X_c^i]^2}{\sigma_{X^i}^2} = \frac{\sigma_{X_c^i}^2}{\sigma_{X^i}^2} \\ \frac{\mu_{Y_c^i}^2}{\sigma_{Y_c^i}^2} &= \frac{(\mu_{X_c^i} - \mu_{X^i})^2}{\sigma_{X_c^i}^2} \end{aligned}$$

D'une manière plus générale la normalisation ne change pas le type de distribution des variables aléatoires. La question de savoir si la normalisation doit être faite avant ou après l'algorithme de sélection ne se pose donc plus.

Plus le score est grand, meilleur est la séparation entre classes au sein de la caractéristique. Cet algorithme est sous optimal car il ne prend pas en compte les effets des autres caractéristiques et ne retire pas la redondance.

Listing 3 – Implémentation de l'algorithme de sélection | Critère de Fisher

```

1 function [idx, score] = FisherScore(trainfeatures)
2     mu_c = cellfun(@mean,trainfeatures,'Uniform',false);
3     var_c = cellfun(@var,trainfeatures,'Uniform',false);
4     inter_class = sum(cell2mat(cellfun(@(x) x.^2, mu_c, 'UniformOutput',false)));
5     intra_class = sum(cell2mat(var_c));
6     score = inter_class ./ intra_class;
7     [~,idx] = sort(score, 'descend');
8 end

```

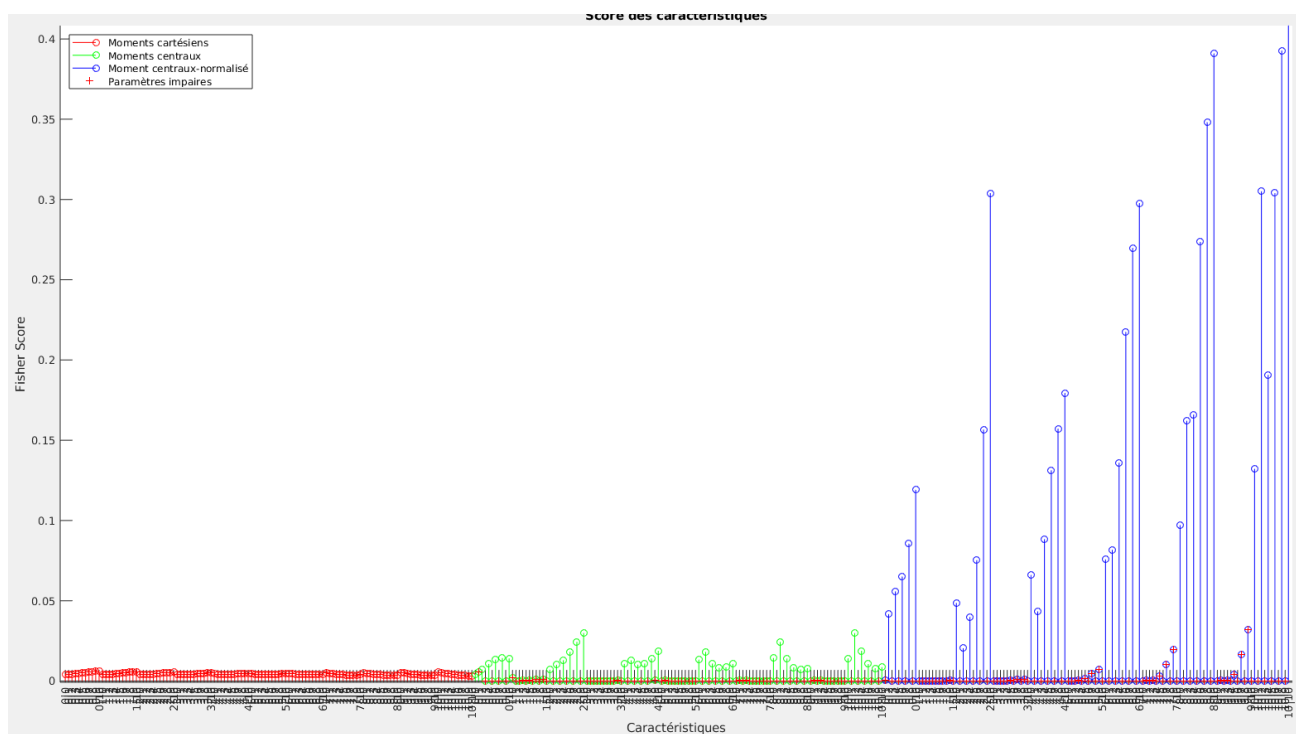


FIGURE 2 – Score des caractéristiques

On remarque que les moments centraux normalisés avec des coefficients pairs ont des scores largement supérieurs.

Recherche d'invariance dans l'espace de représentation

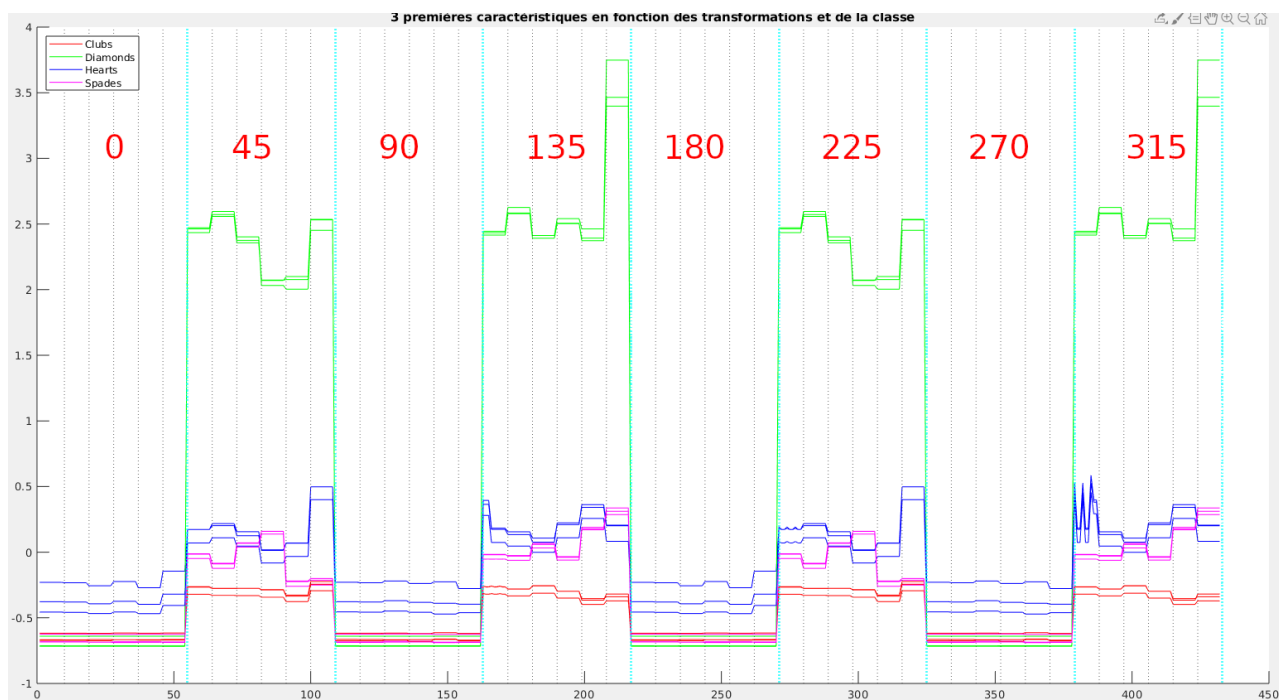


FIGURE 3 – 3 premières caractéristiques en fonction des transformations et de la classe

Le graphique présente les 3 meilleurs caractéristiques selon le critère de Fisher en fonction des transformations appliquer à l'image et de sa classe. Les lignes verticales séparent les transformations géométriques, (**bleu** pour les rotations), (**noire** pour les changements d'échelle).

On remarque tous d'abord que ses 3 caractéristiques sont totalement invariantes à la translation et dans une moindre mesure invariante aux changements d'échelle dans les images d'entraînement. Ces invariances sont valables pour tous les moments centraux normalisés saufs $\eta_{1,0}$ et $\eta_{0,1}$. On peut raisonnablement généraliser cette propriété à l'ensemble de toutes les transformations possibles de l'image.

Les moments centraux normalisés présentent tous de même une certaine régularité à la rotation, les image dont l'angle est séparé de 90 degré semble avoir des moments comparables.

Les deux autres type de moment (Cartésien et Centraux) ne présentent pas ce type d'invariance et ne sont apriorie pas très pertinent.

On remarque aussi que les 3 caractéristiques ont la même allure ils sont en réalité très corrélés ce sont donc des caractéristiques redondantes.

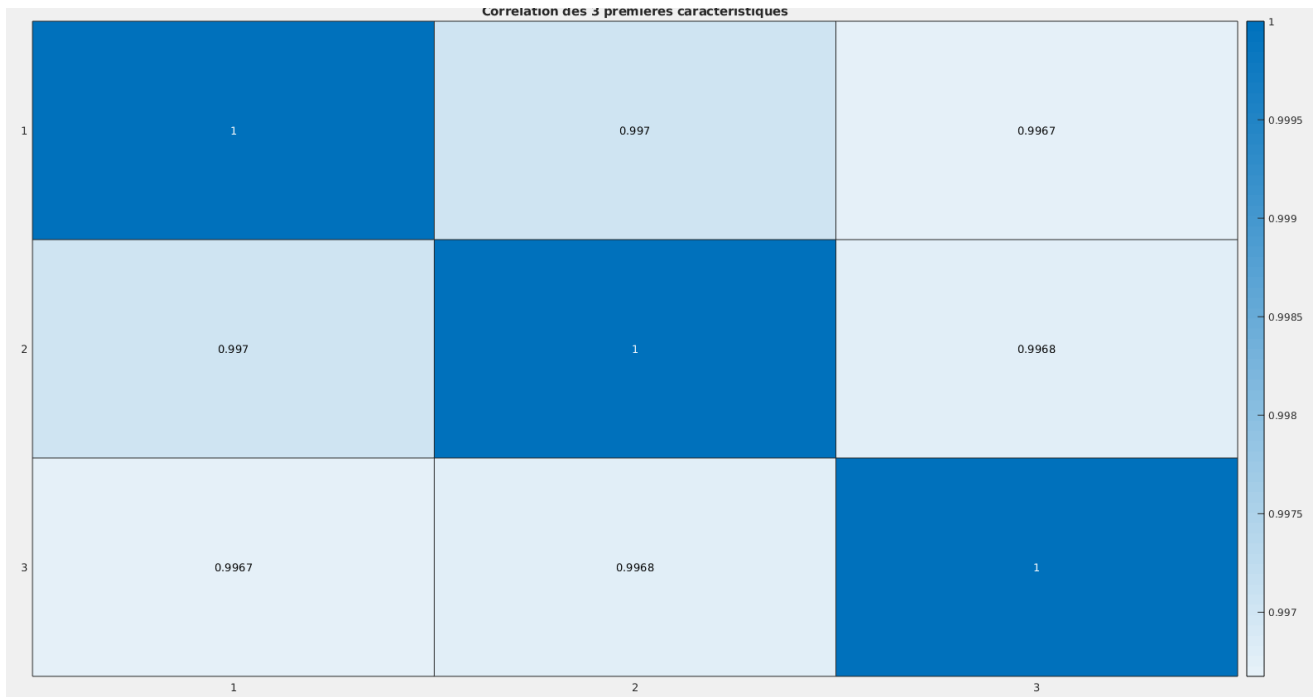


FIGURE 4 – Corrélations des 3 premières caractéristiques

Résumé

Cette première étude nous permet donc de réduire l'espace de représentation en retirant les translations et changement d'échelle dans les images d'entraînement elle permet aussi de retirer les moments cartésiens et centraux. Elle montre aussi que le critère de sélection a tendance à sélectionner des caractéristiques redondantes. On s'assure ainsi une bonne généralisation sur ces types de transformations et nous donne une première idée sur la stratification à adopter.

Recherche de régularité dans l'espace de représentation pour la rotation des images

L'ensemble des images d'entraînement est maintenant constitué uniquement de rotation des images de base.

Labels	Club	Diamonds	Heart	Spade	Commentaire
Rotation	360	360	360	360	$\delta\theta = \frac{\pi}{180}$

De plus l'espace des caractéristiques est généré par les moments centraux normalisés P et Q = 10 privés de $\eta_{1,0}$ et $\eta_{0,1}$ soit un espace de représentation $E_{1440*118}$

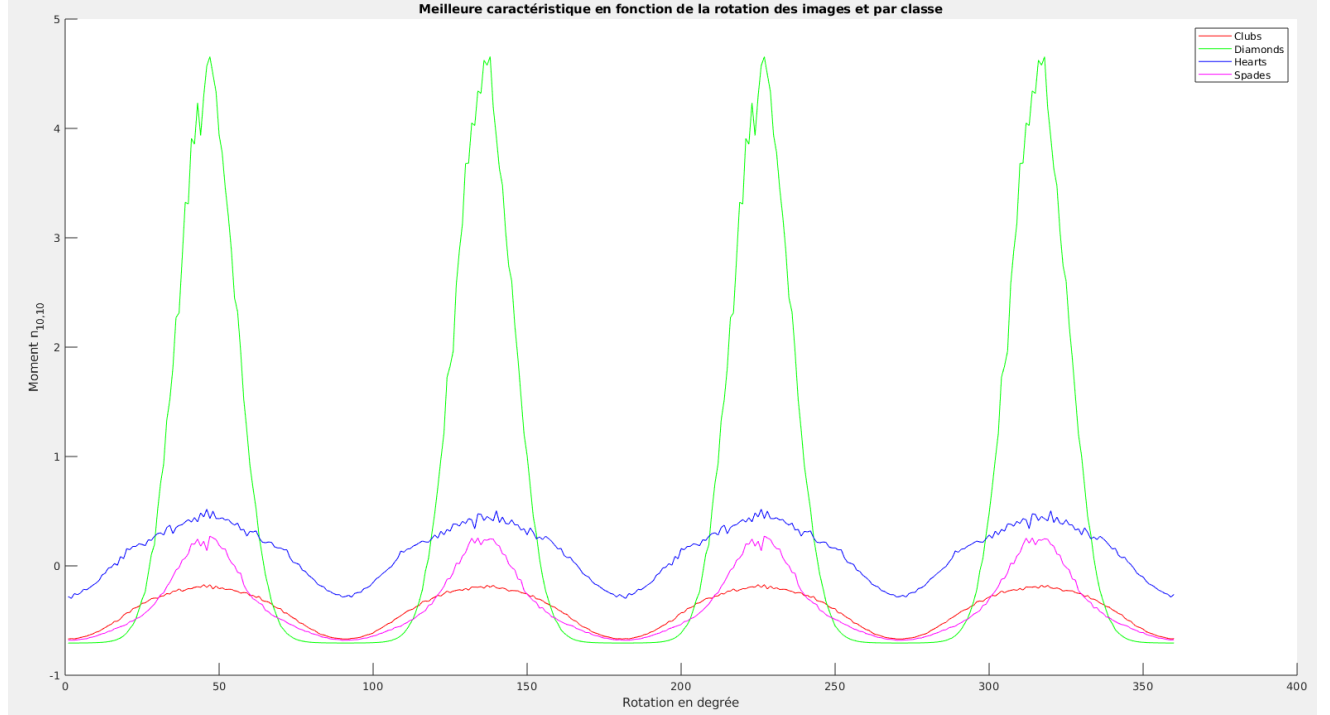


FIGURE 5 – Meilleure caractéristique en fonction de la rotation des images et par classe

On confirme bien une régularité du moment $\eta_{10,10}$ en fonction de la rotation, Cette périodicité des moments en fonction de la rotation de l'image est valable pour l'ensemble des moments centraux normalisés. On peut réduire à présent l'ensemble des images pour des rotations comprises entre 0 et 90 degré.

Recherche du sous-ensemble de caractéristique optimale

L'ensemble des images d'entraînement est maintenant constitué uniquement de rotation des images de base comprise entre $[0, 90]$

Labels	Club	Diamonds	Heart	Spade	Commentaire
Rotation	90	90	90	90	$\delta\theta = \frac{\pi}{180}[0, 90]$

Pour un espace des caractéristiques $E_{364*118}$

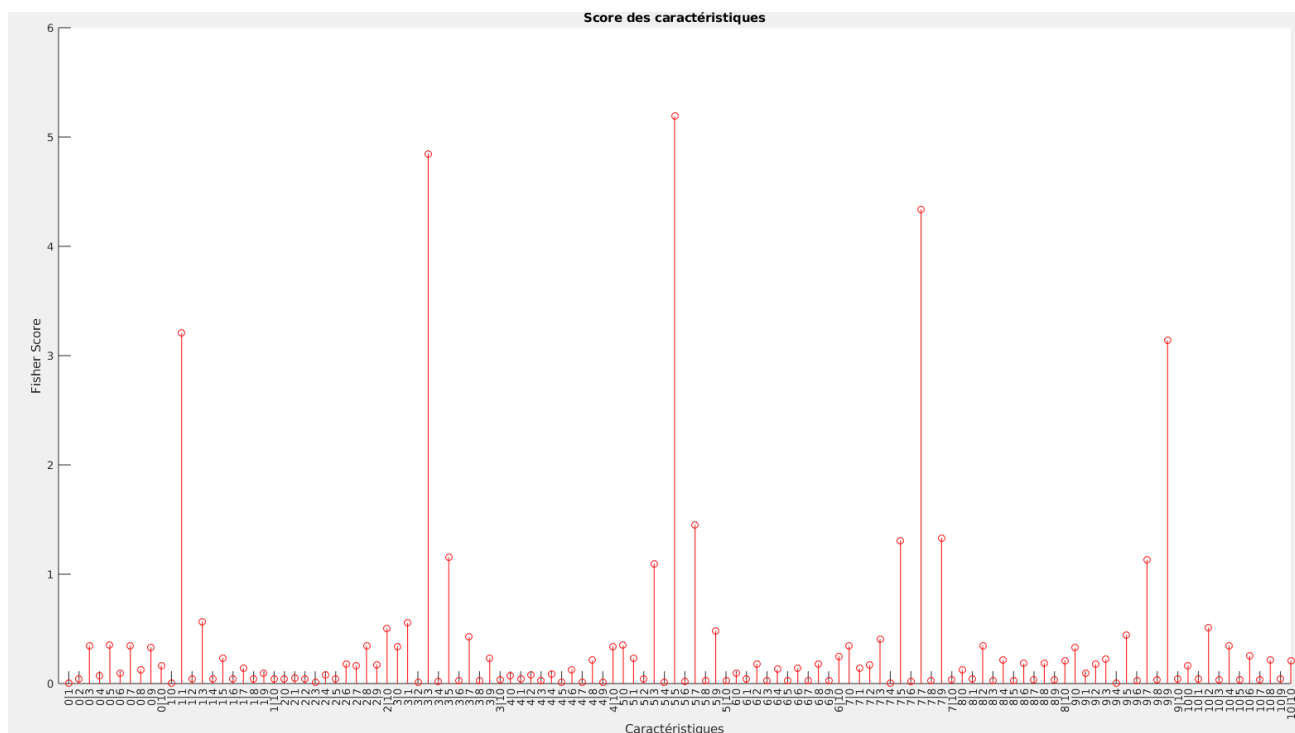


FIGURE 6 – Classement des caractéristiques avec le critère de Fisher

Le critère de fisher nous donne à présent un nouveau classement.

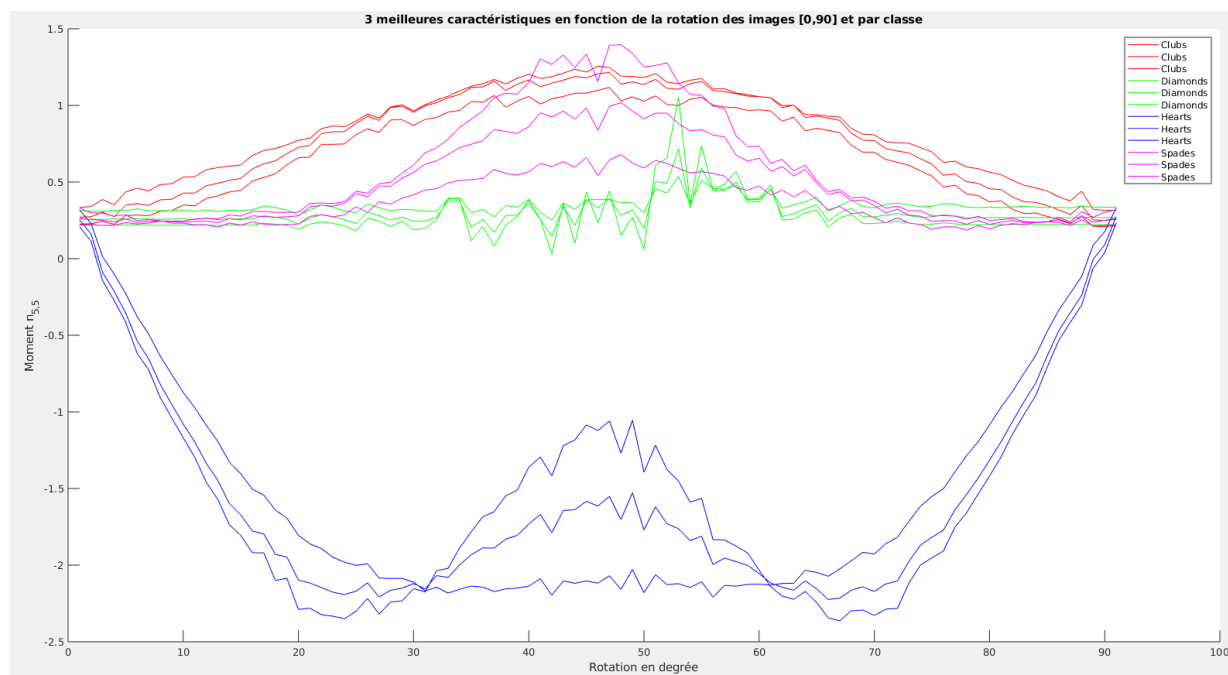


FIGURE 7 – 3 meilleures caractéristiques en fonction de la rotation des images [0 90] et par classe

On remarque tous d'abord que la catégorie "Heart" est bien discriminée sur cette caractéristique mais les 3 autres ne le sont pas.

Il est clair depuis le début que le problème ne va pas se résoudre en 1 seule dimension. Il est temps d'abandonner le critère de fisher qui nous aura tous de même données de bonne intuition au profit d'une approche plus optimale.

Pour sélectionner le sous-ensemble de caractéristique optimal nous allons adopter une approche wrapper, sur un classifieur K-NN avec $k = 1$. Les difficultés évoquées pour stratifier les données d'entraînement et de test sont à présent résolues. Les données d'entraînement sont les images de rotation sur la plage $[0\ 90]$ et les données de tests sont l'ensemble d'entraînement initial – les données de test. Bien qu'on puisse retirer les translations.

Images de tests					
Labels	Club	Diamonds	Heart	Spade	Commentaire
Rotation	8	8	8	8	$\delta\theta = \frac{\pi}{4}$
Zoom	5	5	5	5	Plage $[0.5, 3]/\{1\}\delta z = 0.5$
Translation	9	9	9	9	8 points cardinaux + centre
Images d'entraînement					
Rotation	90	90	90	90	$\delta\theta = \frac{\pi}{180}[0, 90]$

La réduction de la dimensionnalité et des données d'entraînement ne va toujours pas nous permettre une recherche exhaustive car qui impliquerait de tester encore $\binom{N}{118}$. Pour un couple de features cela représente 6903 combinaisons uniques.

Mais il est inutile de tester les combinaisons de caractéristique fortement corrélée car elle n'apporterait que très peu d'informations supplémentaires. Les moments au sein de l'espace ayant la même allure nous pouvons prendre le risque de choisir les couples de caractéristique les plus décorrélés possible. Il reste donc au maximum 118 combinaisons à tester.

Résultats des modèles

On a entraîné 109 modèles, le tableau donne les 5 meilleures performances associées au moment testé.

Feature1	Feature2	Accurate
$\eta_{0,2}$	$\eta_{2,0}$	0.8618
$\eta_{0,4}$	$\eta_{2,0}$	0.8097
$\eta_{6,2}$	$\eta_{0,8}$	0.7806
$\eta_{4,0}$	$\eta_{0,2}$	0.7410
$\eta_{4,10}$	$\eta_{8,0}$	0.7361

Les moments sont tous d'ordre pair, on aurait pu réduire davantage l'espace de représentation en les retirant dès le début comme l'indiquait le critère de Fisher. On a un score de 0.8618 avec la combinaison des moments $\eta_{0,2}$ et $\eta_{2,0}$

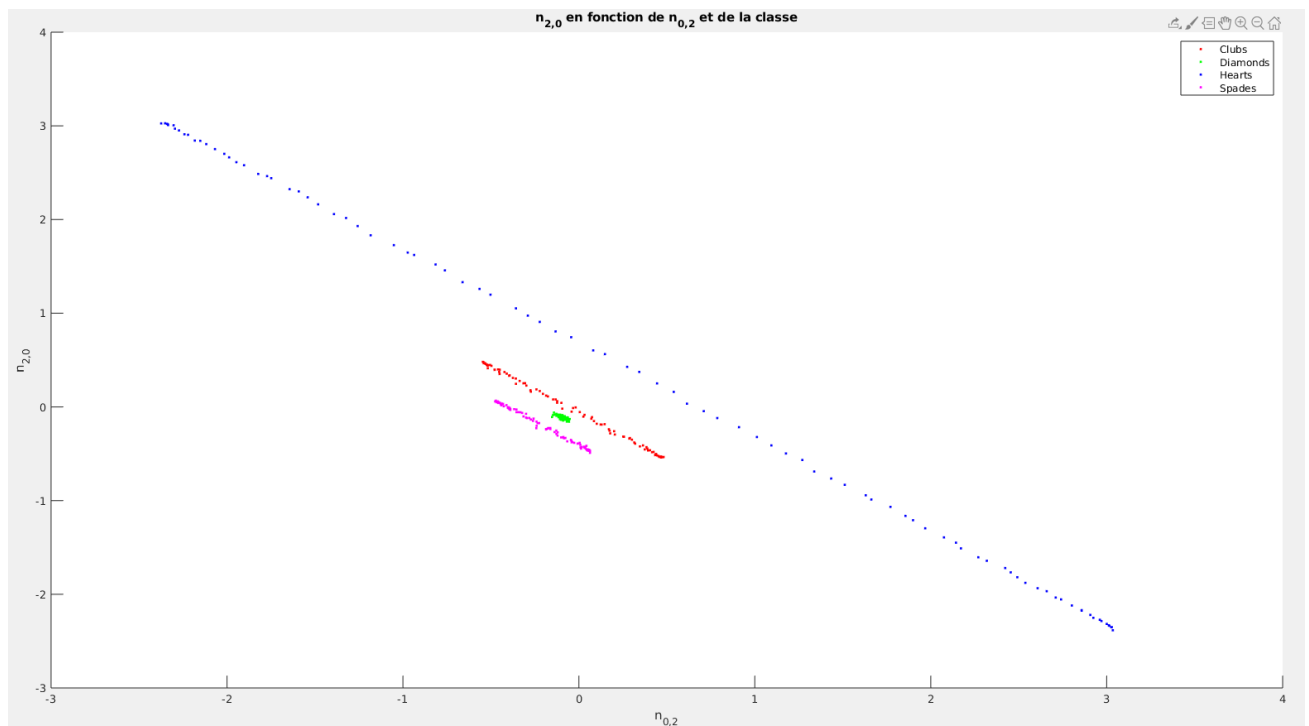


FIGURE 8 – $\eta_{2,0}$ en fonction de $\eta_{0,2}$ et de la classe

Les données dans cet espace sont linéairement séparables on va donc tenter le classifieur SVM avec un noyau linéaire.

Modèle SVM

Un entraînement d'un classifieur SVM avec les deux moment $\eta_{0,2}$ et $\eta_{2,0}$ nous permet d'obtenir ce resultat.

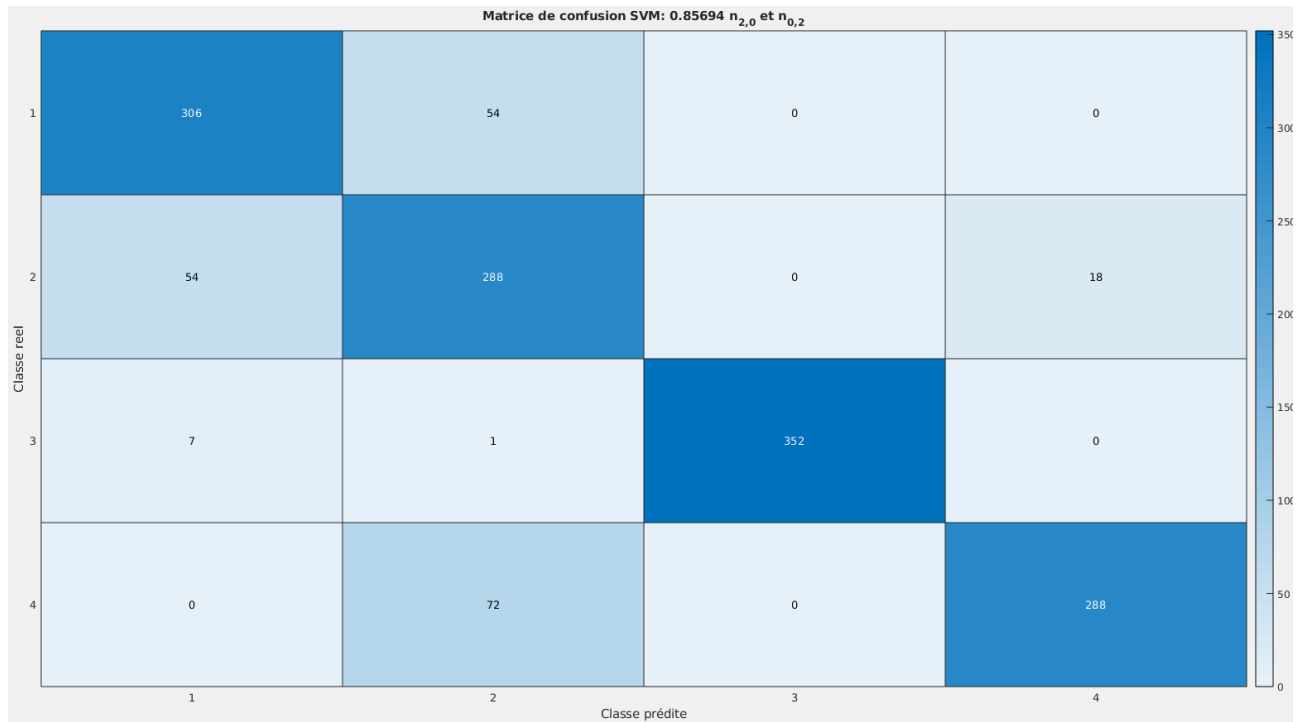


FIGURE 9 – Matrice de confusion du classifieur SVM

Un score de 0.8569 légèrement moins performants que le modèle K-NN mais plus efficients (Gain de mémoire et en temps de calcul).

Mais on peut encore optimiser l'efficacité du modèle à l'aide d'une analyse en composante principale

Analyse en composantes principales

On va s'autoriser à faire des combinaisons linéaires entre les moments.

Toujours sur le même espace de représentation l'analyse en composantes principales nous donne ces coefficients qui correspondent à une rotation de 45 degré.

$$\text{Coefficients} = \begin{bmatrix} 0.7071 & 0.7071 \\ -0.7071 & 0.7071 \end{bmatrix}$$

Calcule du nouveau espace $\{f_1, f_2\}$

$$\begin{bmatrix} f_1 \\ f_2 \end{bmatrix} = \begin{bmatrix} 0.7071 & 0.7071 \\ -0.7071 & 0.7071 \end{bmatrix} * \begin{bmatrix} \eta_{2,0} \\ \eta_{0,2} \end{bmatrix}$$

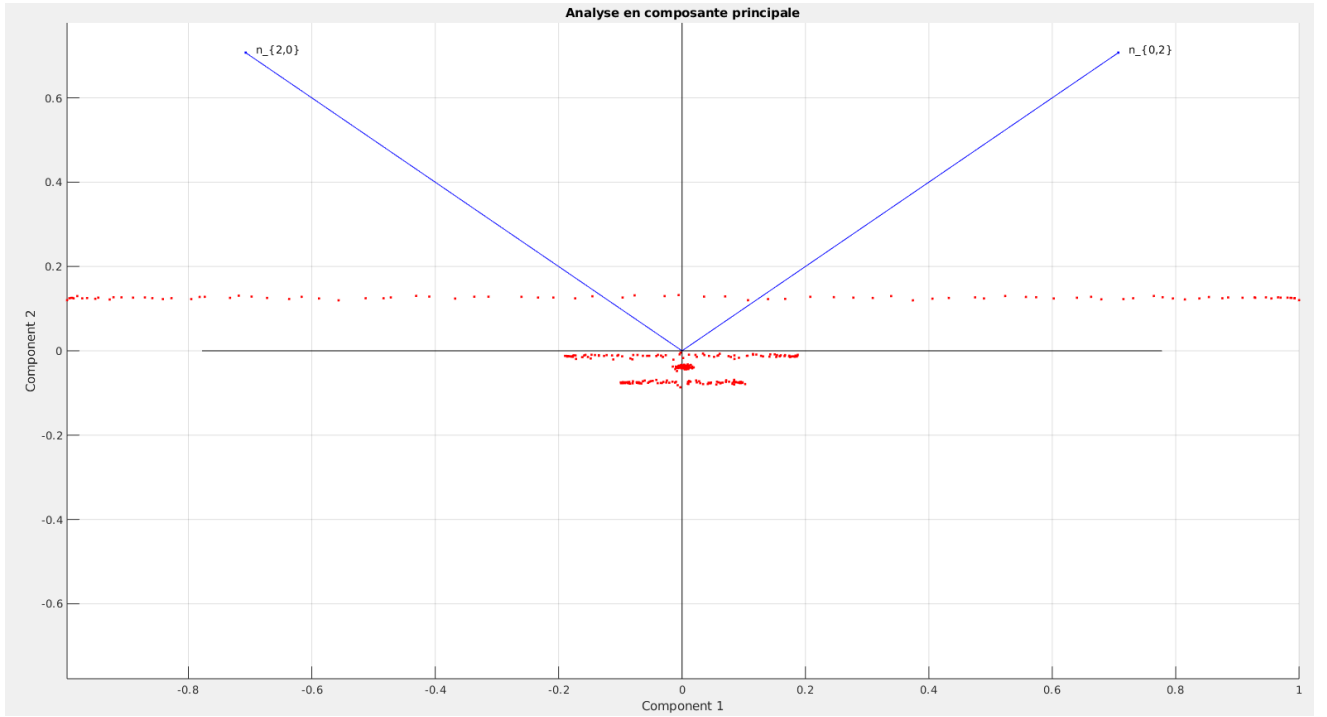


FIGURE 10 – Espace de représentation modifié par l'ACP

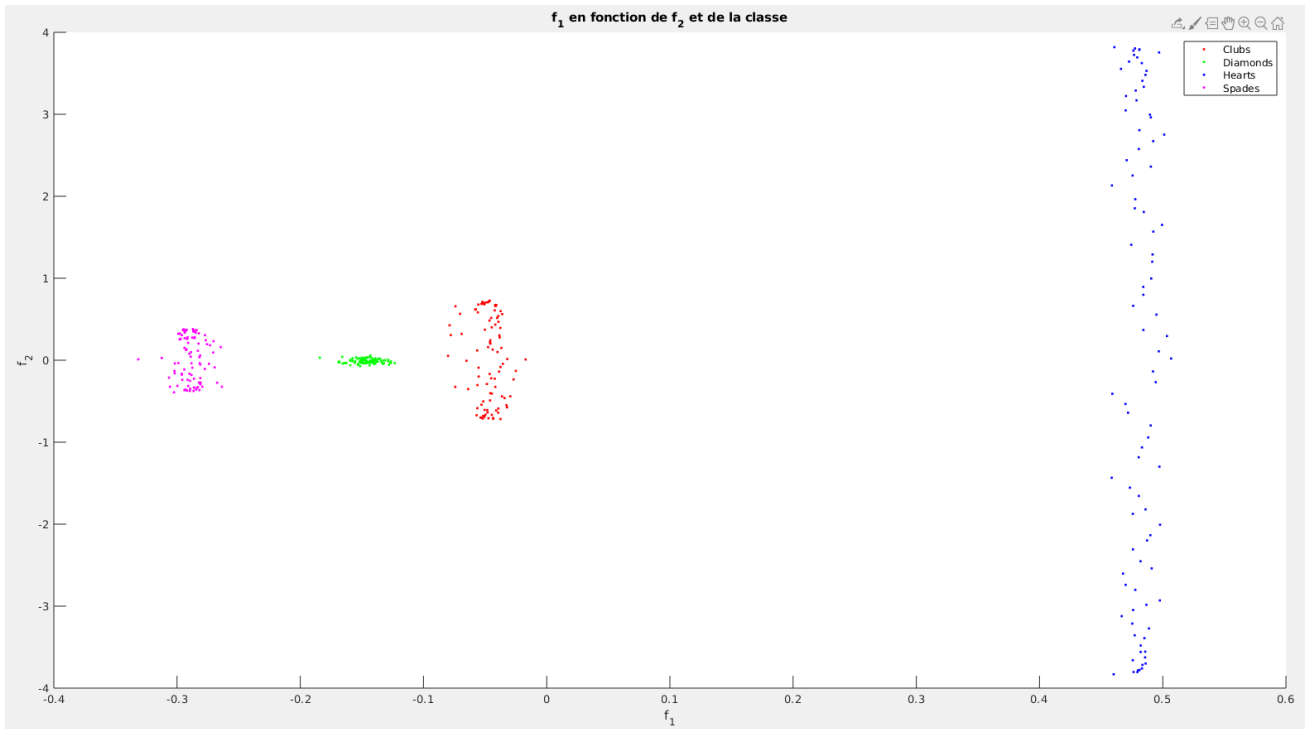


FIGURE 11 – Espace de représentation modifié par l'ACP

f_2 n'étant plus une caractéristique discriminante on peut la retirer il reste donc $f_1 = \eta_{2,0} + \eta_{0,2}$ ce qui correspond au moment de Hu à l'ordre 1 qui est effectivement un moment invariant RTS.

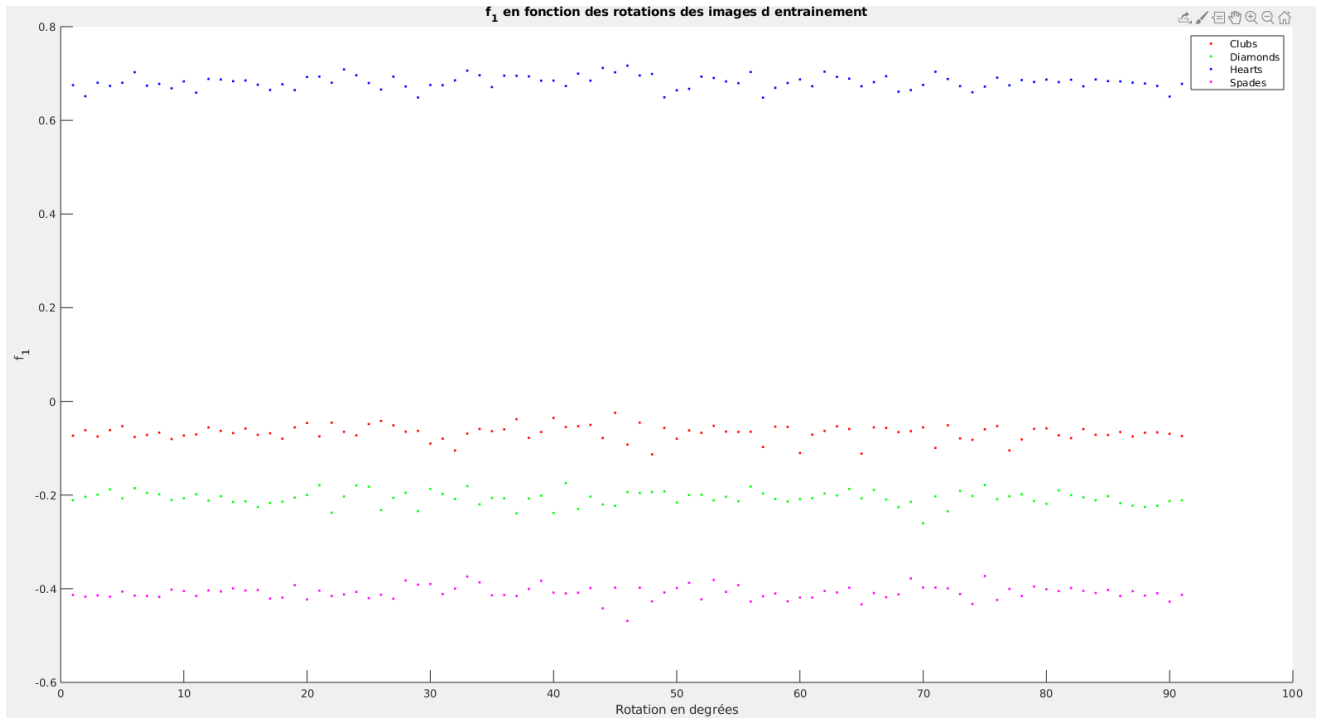


FIGURE 12 – Moment de Hu d'ordre 1 en fonction des rotations des images [0,90]

Modèle SVM, moment de Hu d'ordre 1

Un entraînement d'un classifieur SVM avec le moment de Hu $I_1 = \eta_{0,2} + \eta_{2,0}$ nous permet d'obtenir ce résultat.

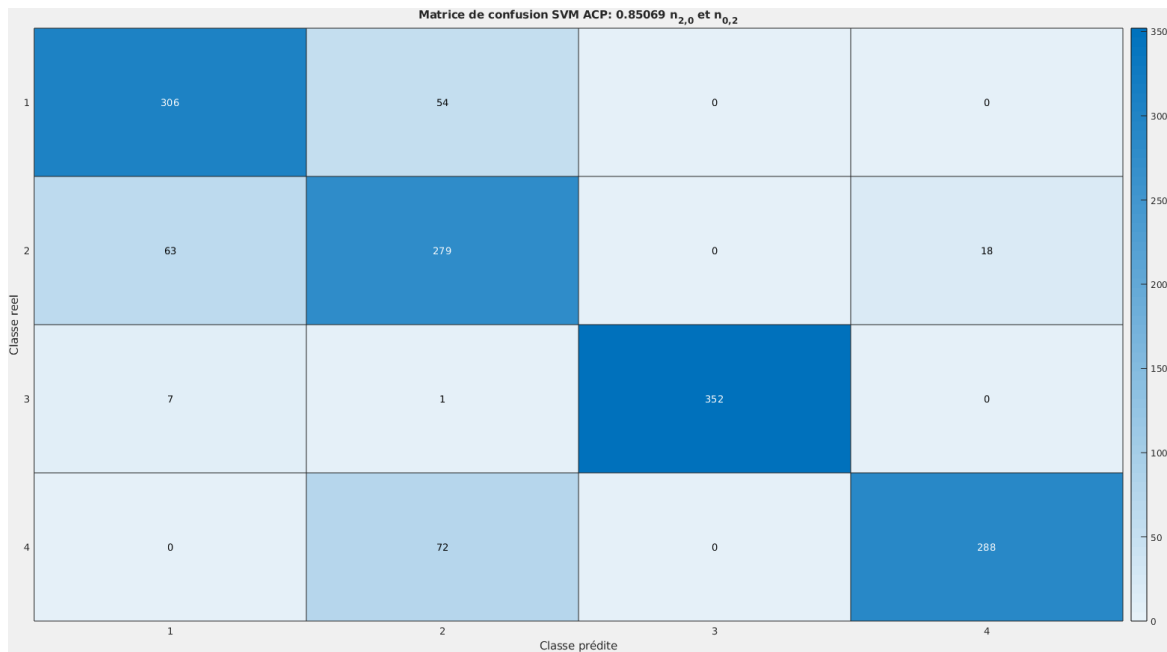


FIGURE 13 – Matrice de confusion du classifieur SVM : Moment de Hu

Ce modèle présente un score de 0.85 ce qui est comparable au modèle précédent mais sépare les classes avec seulement 3 paramètres.

Les sources d'erreurs du modèle

Les données sont "bruitées" par les changements d'échelle des images.
Les erreurs des différents modèles entraînés s'expliquent en grande partie grâce à ce graphique.

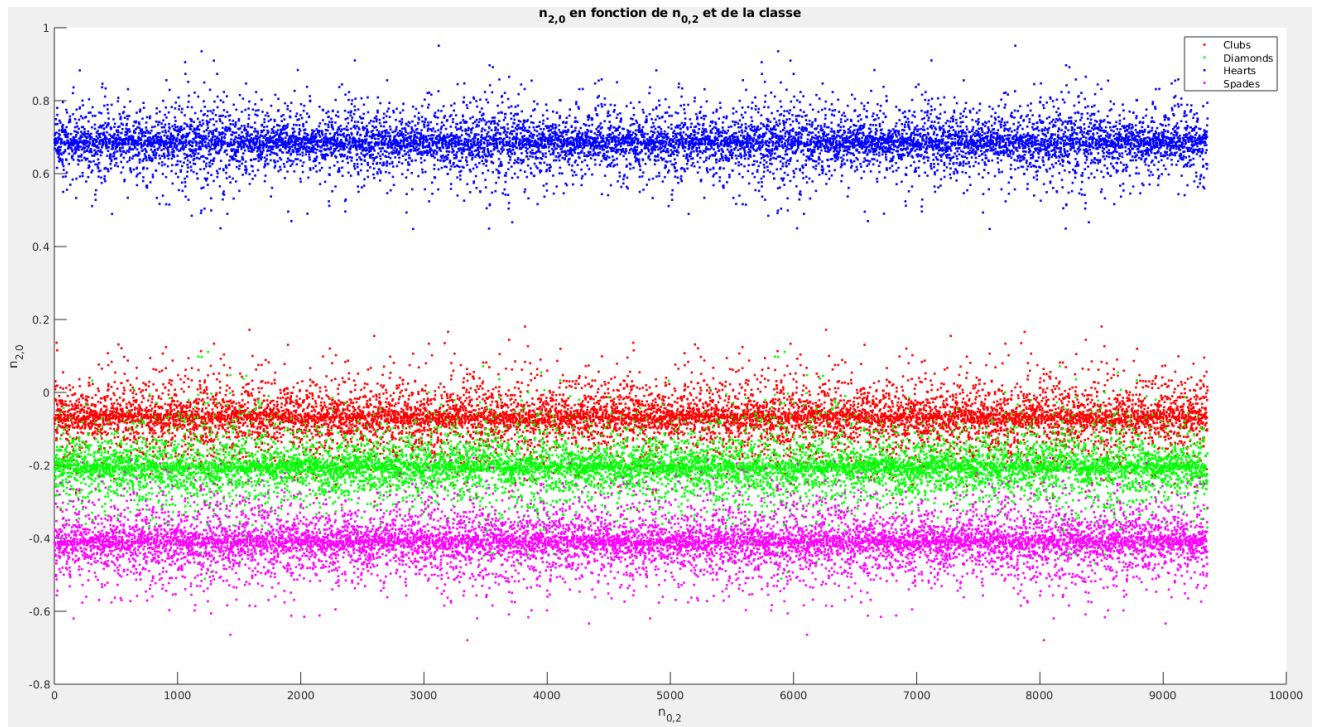


FIGURE 14 – Moment de Hu d'ordre 1 en fonction des transformation de l'image, rotation et changement d'échelle

	Clubs	Diamonds	Hearts	Spades
Classe	1	2	3	4

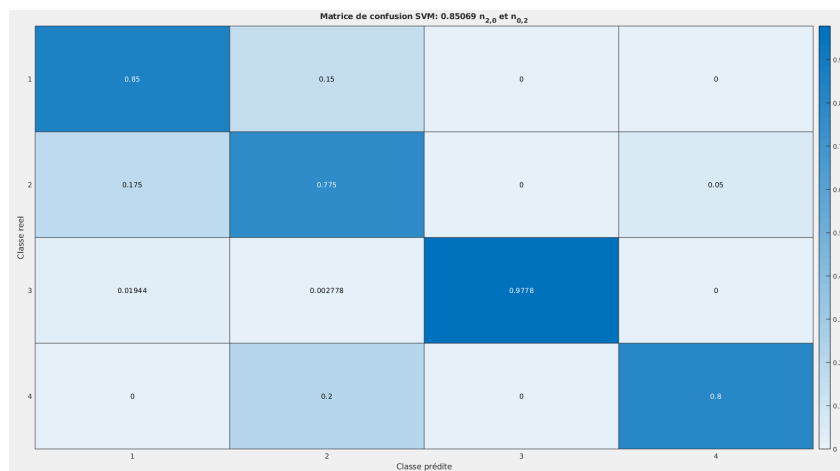


FIGURE 15 – Matrice de confusion normalisé du classifieur SVM : Moment de Hu

Les taux d'erreur importants de la classe 2 avec les classes 1 et 4 est expliqués par la petite marge entre ces 3 distributions.

Maximum a posteriori

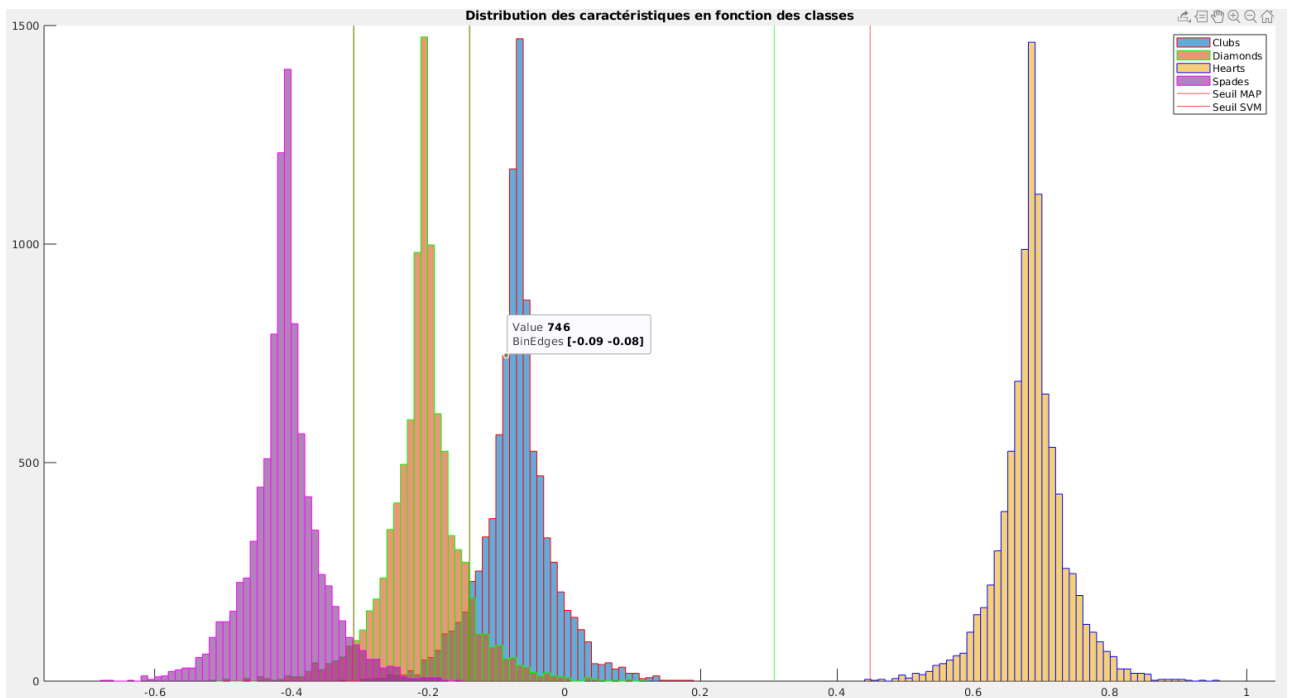


FIGURE 16 – Distribution des caractéristiques en fonction des classes

Le modèle SVM n'étant pas entraîné avec ces bruits il n'a pas pu définir des seuils optimaux, les seuils MAP vont permettre d'améliorer légèrement la performance à 0.8756.

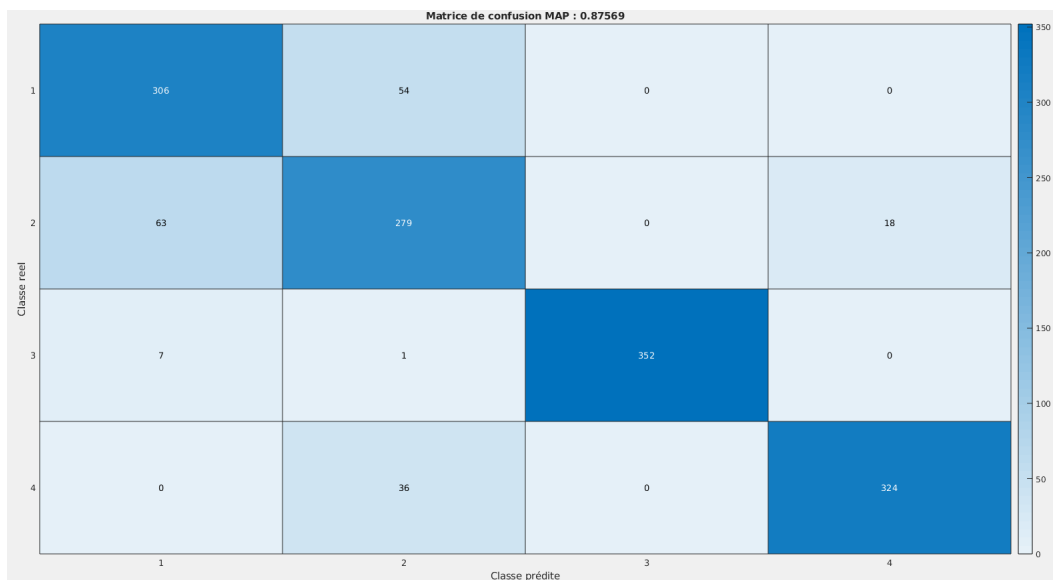


FIGURE 17 – Matrice de confusion du classifieur MAP : Moment de Hu

En fait il réduit les faux négatifs de la classe 4(P) avec la classe 2(N).

Conclusion

En conclusion nous avons réussie à réduire la dimensionnalité de l'espace des features à 2 dimension. A partir de cela nous avons entraîné ces modèles :

Modèle	Performance	Clubs	Diamonds	Hearts	Spades	Nombre de paramètres
K-NN	0.8618	0.9	0.7	0.9972	0.85	2880
SVM	0.8569	0.85	0.8	0.9778	0.8	8
SVM-HU	0.8506	0.85	0.775	0.9778	0.8	3
MAP	0.8756	0.85	0.775	0.9778	0.9	3

Grâce à la méthodologie nous avons pu stratifier correctement les données de tests et d'entraînement, ce qui permet de fournir une mesure de performance plus fiable. Le meilleur modèle permet une prédiction de 87.56 % il sera difficile de faire mieux à cause des changements d'échelle des images, il existe peut-être un sous-ensemble à plus haute dimension capable d'être plus performant, on peut aussi peut-être envisager un prétraitement sur les images pour réduire cette variance.

Quelques liens

Présentation des différentes approches et des problématiques de sélection de caractéristiques.

Sélection de caractéristiques:méthodes et applications

Présentent 22 méthodes d'algorithmes de sélection de caractéristiques par une approche Filter

Benchmark for filter methods for feature selection in high-dimensional classification data