

L'ensemble du projet (sujet, données, codes et rapport) est disponible sur :
<https://github.com/AspirantMoutarde/ISAR-Image-Classification>.

Partie 1 : Analyse de données ISAR

Question 1 : Analyse du jeu de données

Le jeu de données contient 810 éléments, répartis en 5 classes. Chacune des classes contient 162 éléments. La valeur minimale attendue est 0 et la valeur maximale atteinte est 255. L'espace des données est donc de dimension 256x256. Afin de réduire cet espace nous choisissons un carré de 50x50 au centre de l'image d'origine.

```
Nombre total d'éléments = 810
Nombre de classes : 5
Nombre d'éléments labélisés A10 : = 162
Nombre d'éléments labélisés F4 : = 162
Nombre d'éléments labélisés F14 : = 162
Nombre d'éléments labélisés F15: = 162
Nombre d'éléments labélisés F16 : = 162
Valeur minimale atteinte : 0
Valeur maximale atteinte : 255
```

Figure 1: Analyse du dataset.

Nb : Ces valeurs sont disponibles en appelant la fonction `datasetAnalyse` du script `affichage`.

Question 2 :

La fonction `load_bdd` est a retrouve dans le script `dataLoader` du projet.

Partie 2 : Extraction des caractéristiques discriminantes

Image polaire

Question 1 :

Pour cette question les résultats ont été produits à partir de la première image du jeu de données.

Sur la figure précédente on remarque qu'une rotation de l'image d'origine provoque sa translation dans le plan polaire. En ce qui concerne les changements d'échelle, on remarque qu'un

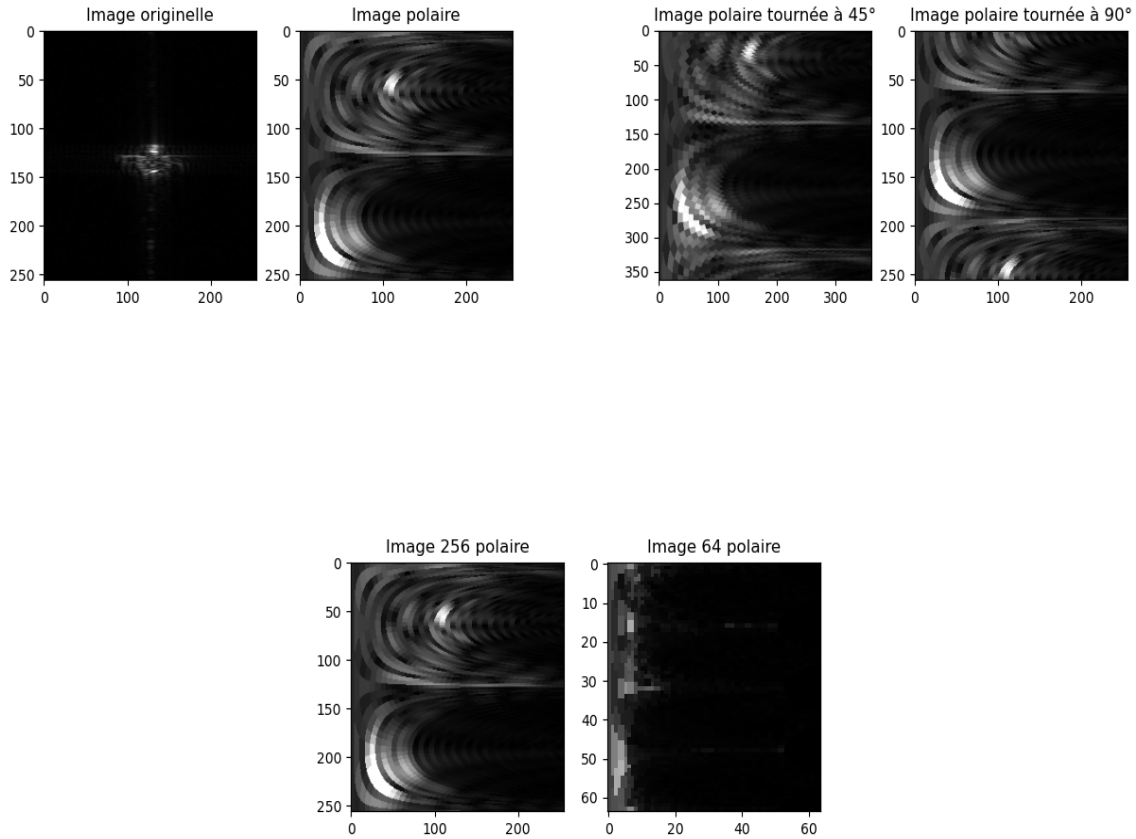


Figure 2: Première image du jeu de données, sa transformation polaire, sa transformation polaire après une rotation de 45° et 90° . Image polaire après changements d'échelle.

agrandissement de l'échelle étire la représentation polaire et qu'une diminution de l'échelle provoque un écrasement de la représentation polaire.

Question 2 :

On observe ici que la transformation polaire ainsi que les composantes I_r et I_θ vont nous permettre d'effectuer la classification de façon convenable.

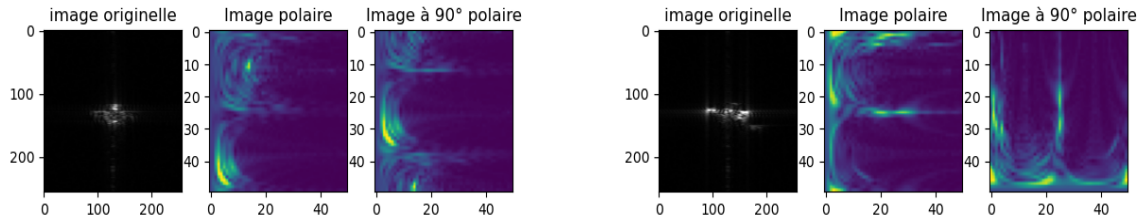


Figure 3: Transformations issues de la seconde transformation polaire.

Partie 3 : Reconnaissance

Question 1 :

Voir descripteurs.py pour le détail de la fonction.

Question 2 :

La séparation des ensembles d'entraînement et de test est effectué avec sklearn dans main.py et la gestion des Knn est effectuée dans reconnaissance.py dont les fonctions sont appelés dans le main.

Question 3 :

Classification report pour K = 1					Classification report pour K = 3					Classification report pour K = 5				
	precision	recall	f1-score	support		precision	recall	f1-score	support		precision	recall	f1-score	support
0	0.98	1.00	0.99	54	0	0.98	1.00	0.99	54	0	0.96	0.98	0.97	54
1	1.00	1.00	1.00	42	1	0.98	1.00	0.99	42	1	0.95	0.98	0.96	42
2	1.00	1.00	1.00	53	2	1.00	1.00	1.00	53	2	0.93	1.00	0.96	53
3	1.00	1.00	1.00	64	3	1.00	0.98	0.99	64	3	1.00	0.95	0.98	64
4	1.00	0.98	0.99	55	4	1.00	0.98	0.99	55	4	0.98	0.93	0.95	55
accuracy			1.00	268	accuracy			0.99	268	accuracy			0.97	268
macro avg	1.00	1.00	1.00	268	macro avg	0.99	0.99	0.99	268	macro avg	0.97	0.97	0.97	268
weighted avg	1.00	1.00	1.00	268	weighted avg	0.99	0.99	0.99	268	weighted avg	0.97	0.97	0.97	268

Figure 4: Rapports sur C1 avec k=1, k=3 et k=5.

Je suis un peu suspicieux au sujet des résultats donnés par $k=1$. Les résultats pour $k=3$ ET $K=5$ sont similaires et très bons, ce qui est cohérent avec le fait que le jeu de données est très bien maîtrisé (i.e. propre). Enfin, dans les trois cas les temps de calculs sont similaires.

Question 4 :

Les résultats semblent similaires avec I_r et I_θ .

Question 5 :

Classification report pour K = 1					Classification report pour K = 3					Classification report pour K = 5				
	precision	recall	f1-score	support		precision	recall	f1-score	support		precision	recall	f1-score	support
0	1.00	1.00	1.00	54	0	0.96	1.00	0.98	54	0	1.00	1.00	1.00	54
1	1.00	1.00	1.00	42	1	0.98	1.00	0.99	42	1	1.00	1.00	1.00	42
2	1.00	1.00	1.00	53	2	0.98	0.98	0.98	53	2	0.95	1.00	0.97	53
3	1.00	1.00	1.00	64	3	1.00	0.97	0.98	64	3	1.00	0.97	0.98	64
4	1.00	1.00	1.00	55	4	1.00	0.98	0.99	55	4	1.00	0.98	0.99	55
accuracy			1.00	268	accuracy			0.99	268	accuracy			0.99	268
macro avg	1.00	1.00	1.00	268	macro avg	0.98	0.99	0.99	268	macro avg	0.99	0.99	0.99	268
weighted avg	1.00	1.00	1.00	268	weighted avg	0.99	0.99	0.99	268	weighted avg	0.99	0.99	0.99	268

Figure 5: Rapports sur C2 avec $k=1$, $k=3$ et $k=5$.

Je suis un peu suspicieux au sujet des résultats donnés par $k=1$. Les résultats pour $k=3$ ET $K=5$ sont similaires et très bons, ce qui est cohérent avec le fait que le jeu de données est très bien maîtrisé (i.e. propre). Enfin, dans les trois cas les temps de calculs sont similaires. Néanmoins, ici on note que C2 est relativement plus rapide que C1.