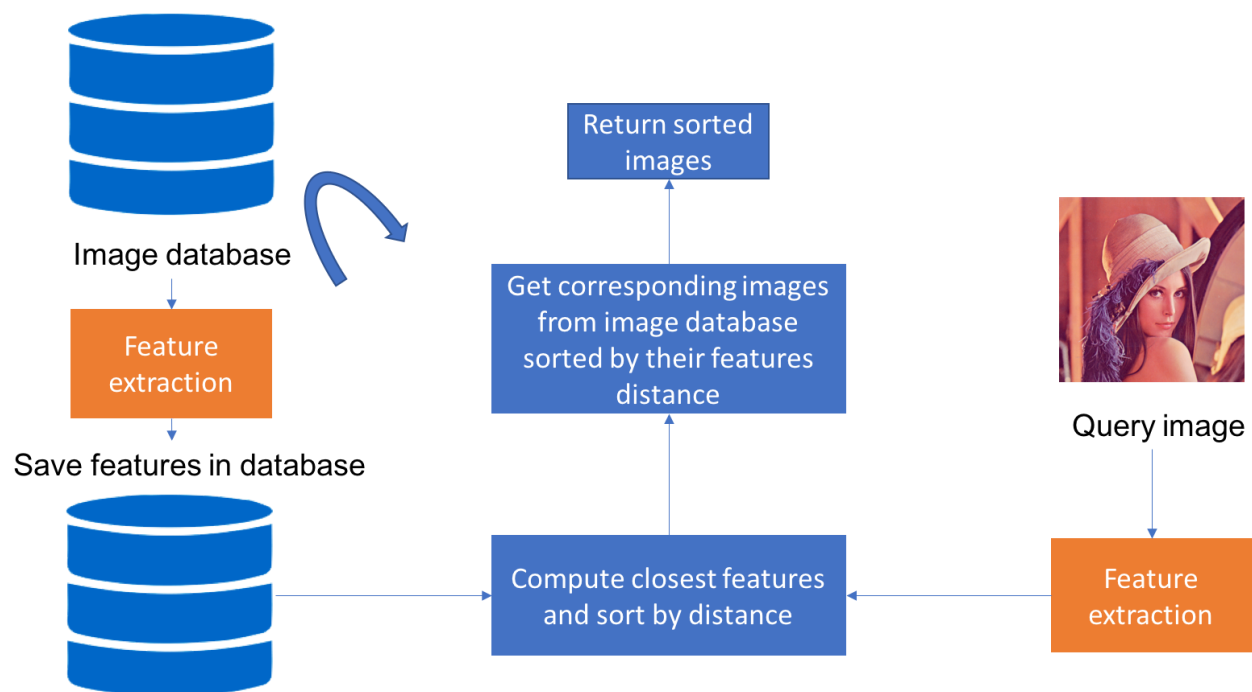


Objectif

La [recherche d'images basée sur le contenu](#) (CBIR) est une méthode permettant de rechercher des images à partir des caractéristiques visuelles, c'est-à-dire induites par les pixels. L'utilisation la plus connue de cette méthode est la fonction de recherche d'images du moteur de recherche Google. Elle s'effectue en quatre étapes :

1. Extraction des caractéristiques d'une base de données d'images pour former une base de données des caractéristiques,
2. Extraction des caractéristiques de l'image fournie en entrée,
3. Recherche des caractéristiques les plus similaires dans la base de données,
4. Retour de l'image associée aux caractéristiques trouvées.



Cette méthode repose sur l'algorithme d'extraction et la comparaison des caractéristiques.

Mesure de similarité

- [Similarité cosinus](#)
- [Distance de manhattan](#)
- [Distance euclidienne](#)

Algorithme d'extraction

- [AKAZE](#)
- [ORB](#)
- [VGG16](#)
- [NasNet](#)
- [EfficientNet](#)

L'objectif de trouver la bonne combinaison (algorithme d'extraction & mesure de similarité) qui permet d'avoir des réponses pertinentes.

Jeux de données

Nous avons utilisé le jeu de donnée [Apparel](#) pour notre évaluation.

Elle contient 11385 images réparties en 37 catégories, divisé en deux jeux :- jeu d'entraînement: 10833 images- jeu de test: 5337

Evaluation

Diversité de des données

L'ensemble des algorithmes d'extraction a reçu le même nombre d'images pour constituer les bases de données. Néanmoins, certains n'ont pas pu extraire les données de toutes les images.

AKAZE	ORB	VGG	NasNet	EfficientNet
10809	10832	10833	10833	10833

Les algorithmes basés sur des réseaux neuronaux n'ont pas eu de mal à extraire toutes les données, contrairement aux algorithmes basés sur les descripteurs, particulièrement AKAZE.

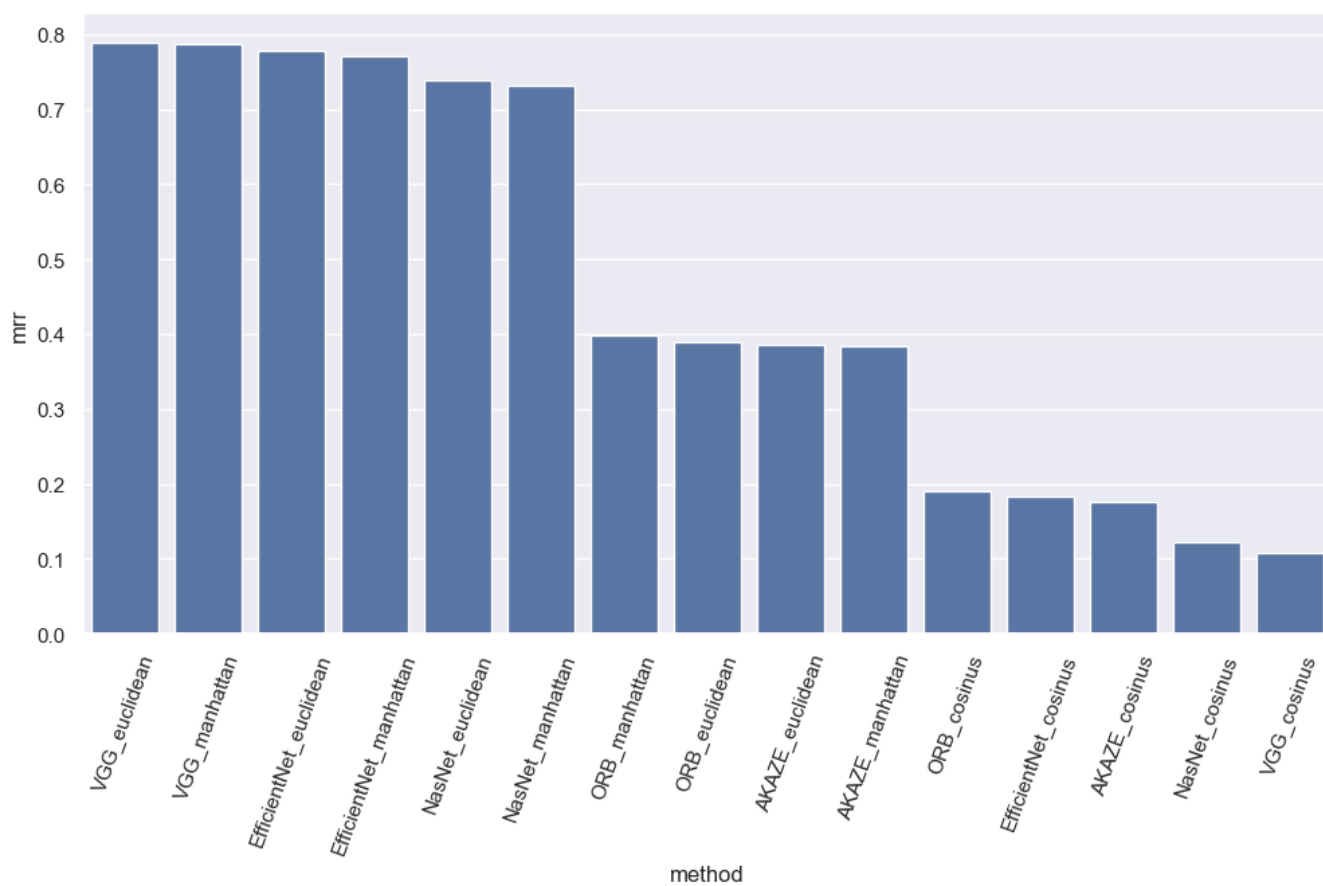
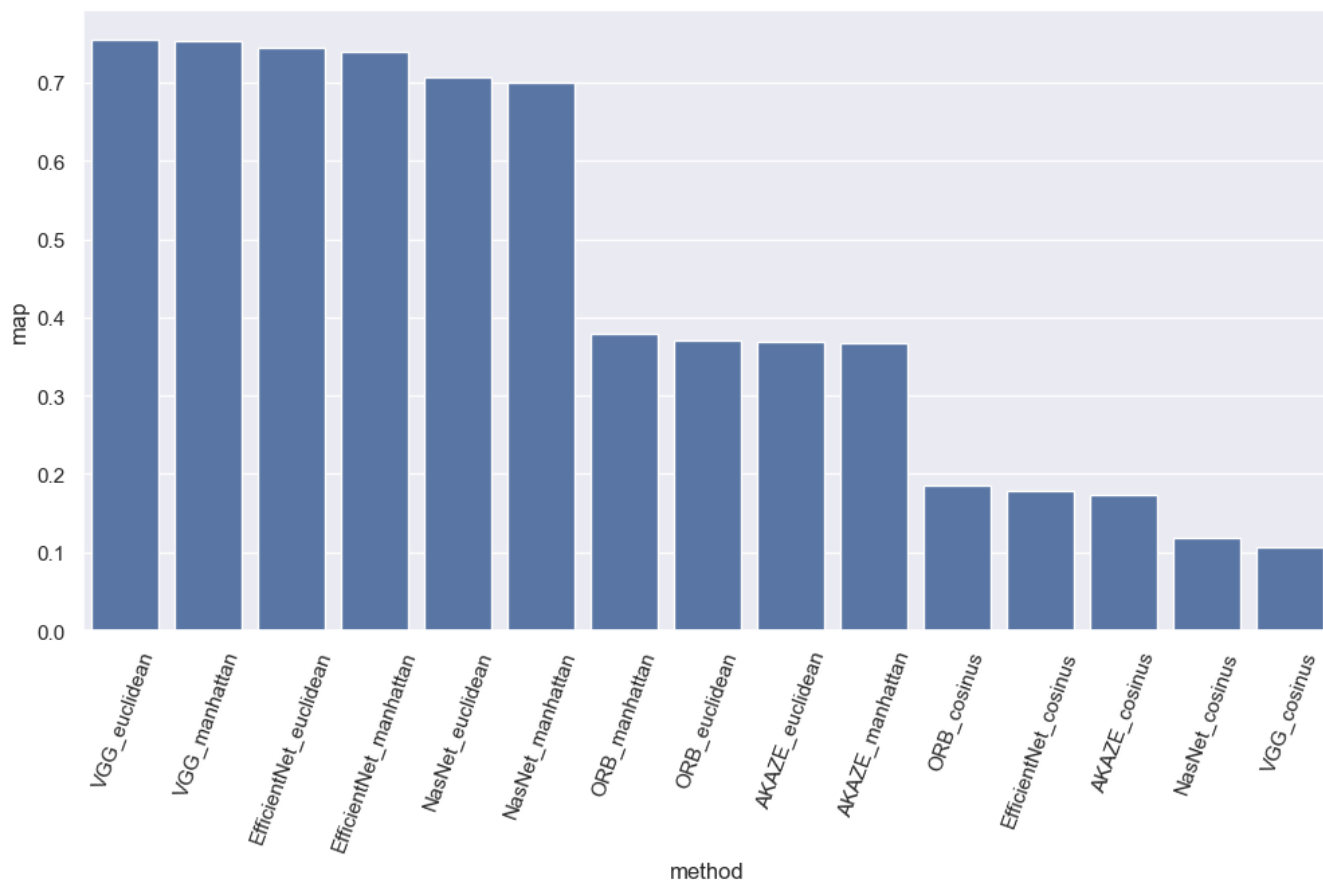
Métriques d'évaluation

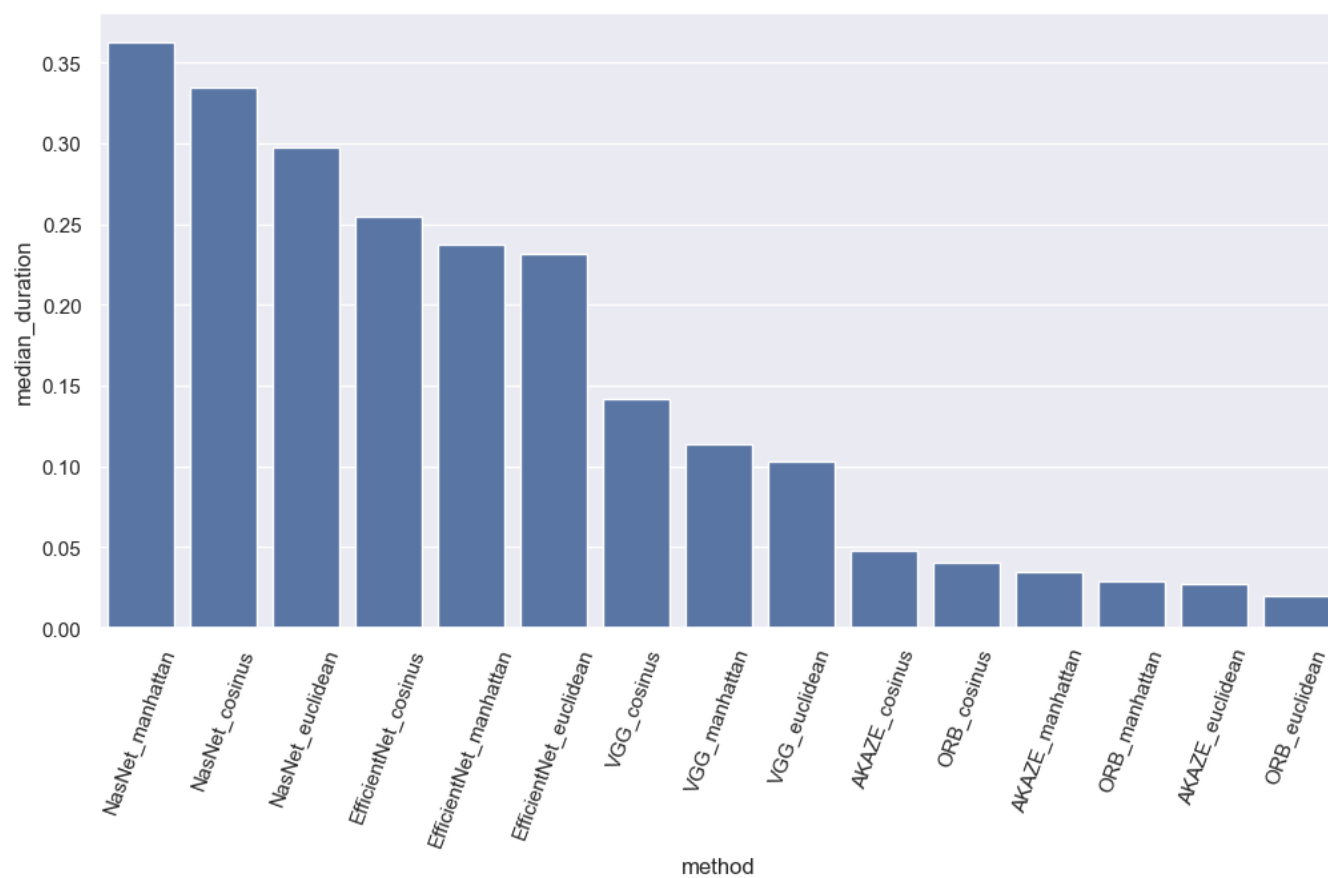
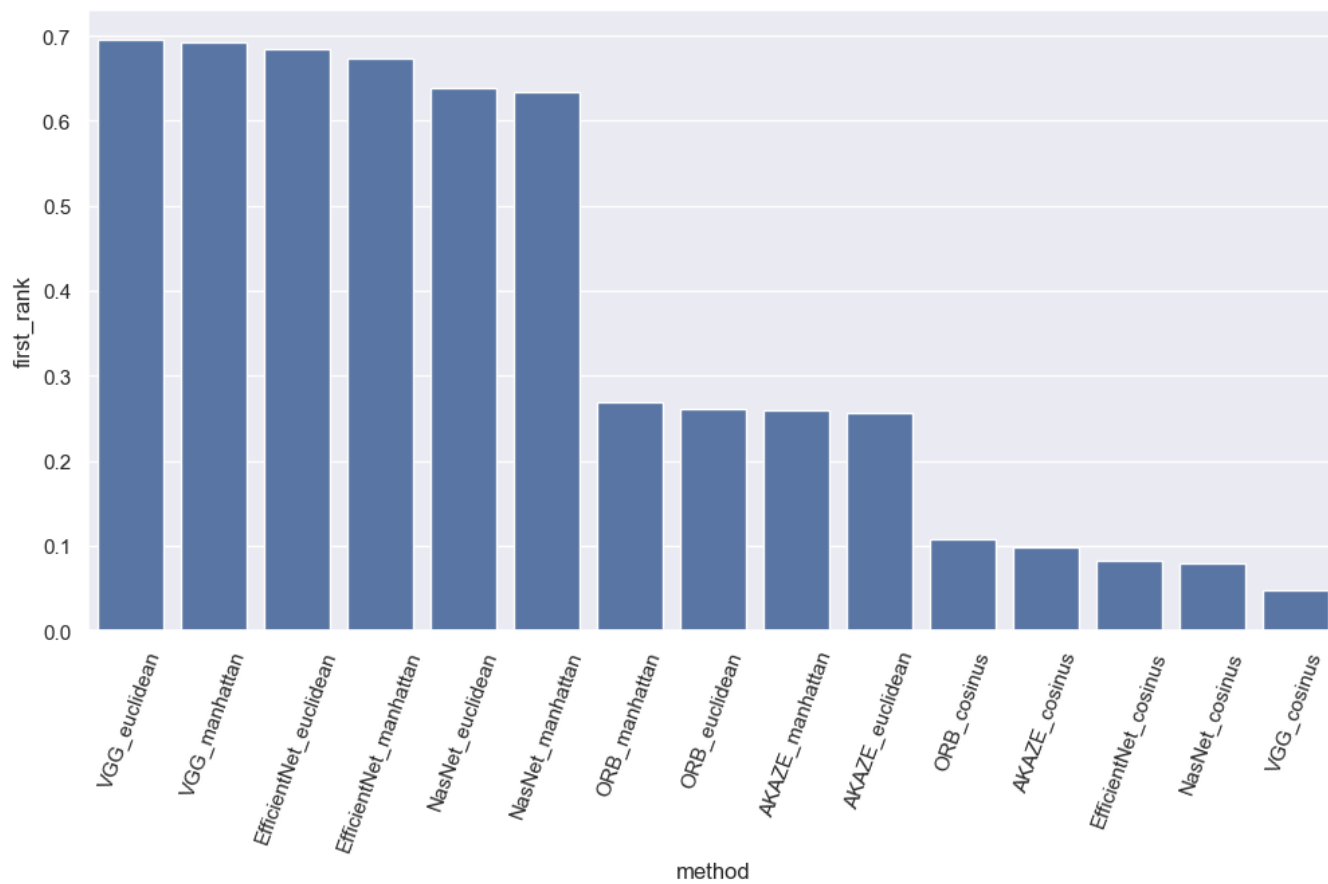
- Mean Average Precision (MAP) pour mesurer la robustesse du système
- Mean Reciprocal Rank (MRR) pour mesurer la pertinence du premier élément
- Temps de traitement d'une requête

Pour plus de détail [voir](#)

Colors

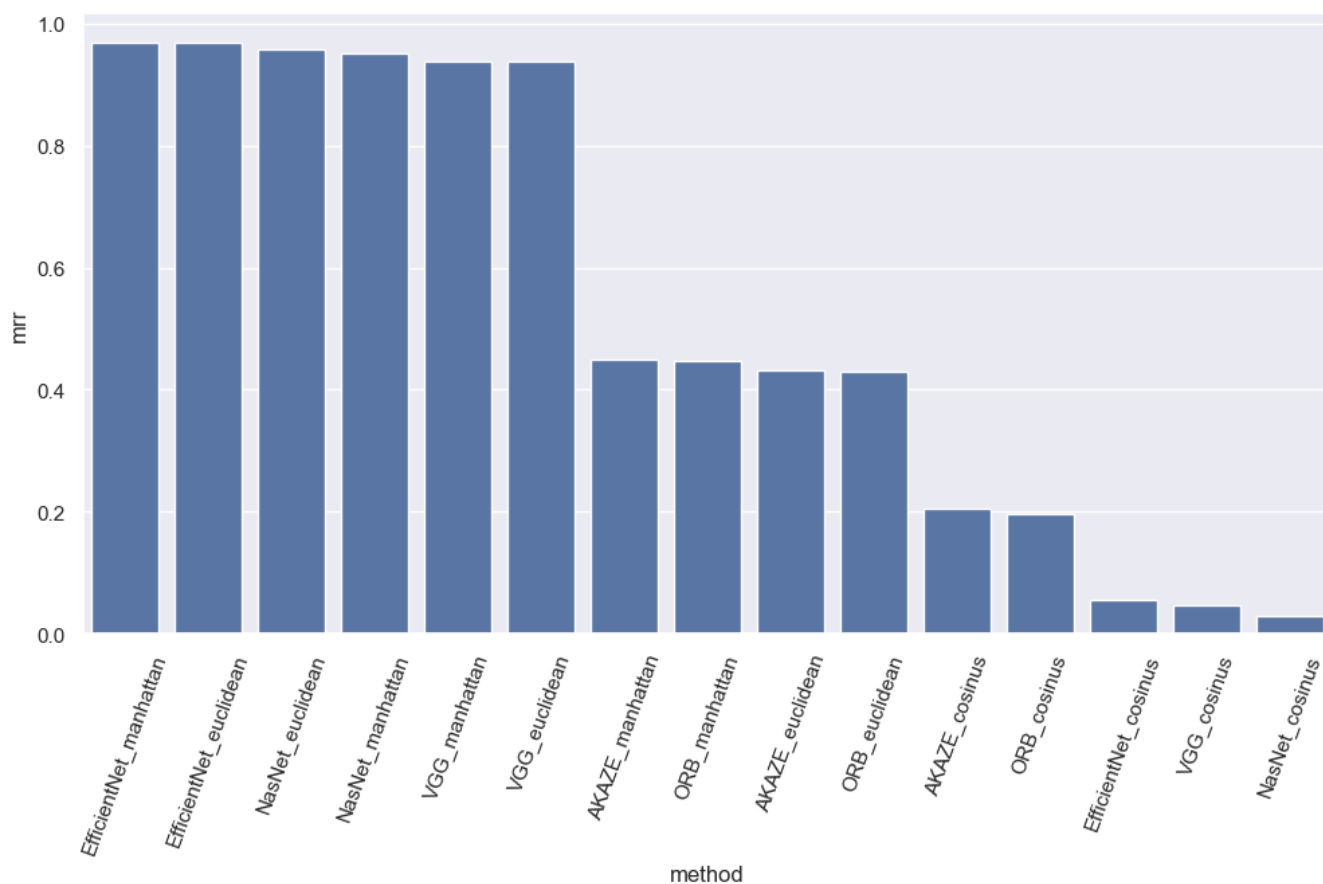
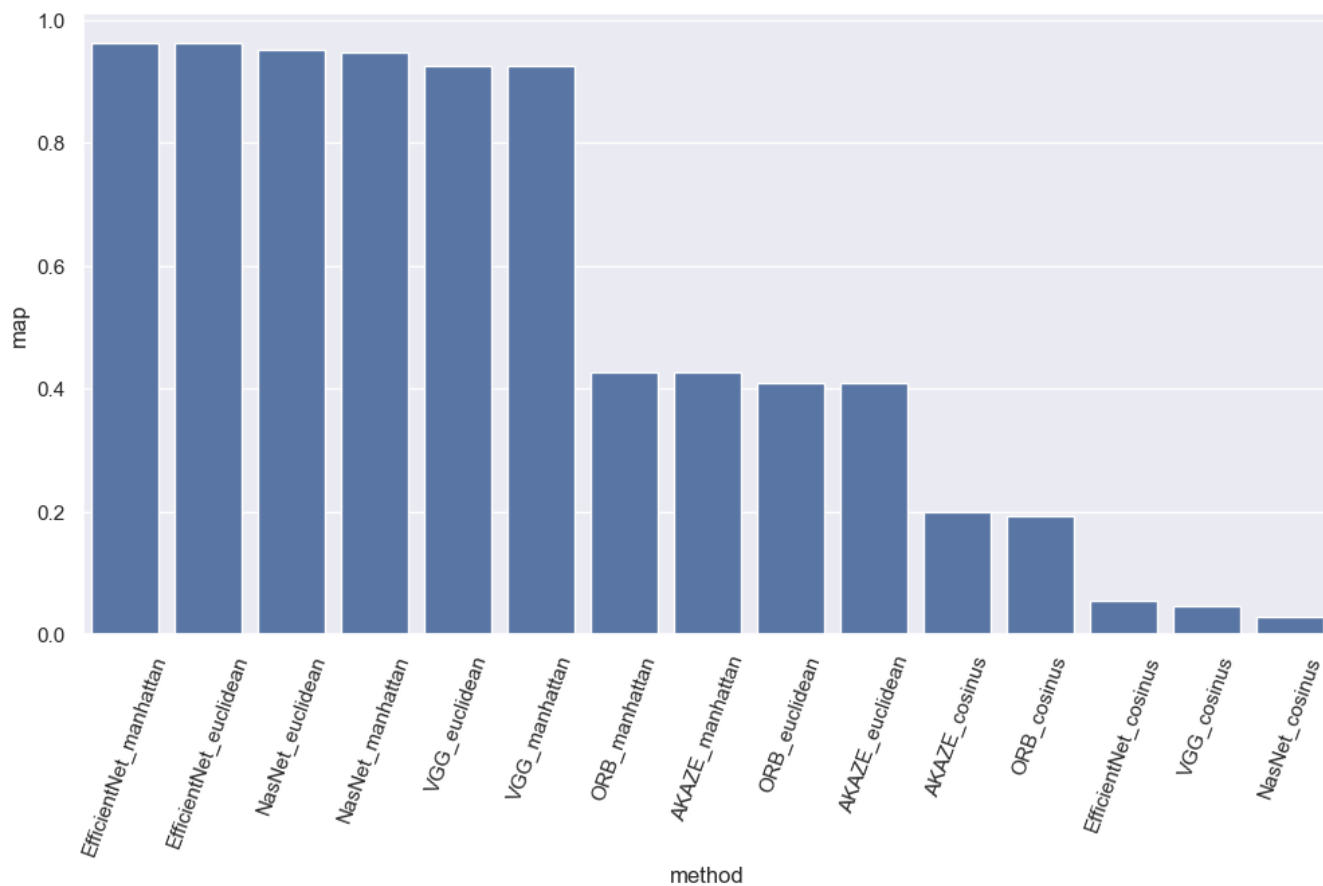
method	map	mrr	first_rank	median_duration
AKAZE_cosinus	0.173069	0.175045	0.097433	0.04769
AKAZE_euclidean	0.369149	0.385647	0.256511	0.027107
AKAZE_manhattan	0.367451	0.384183	0.259134	0.034335
ORB_cosinus	0.185993	0.189991	0.107176	0.040173
ORB_euclidean	0.370821	0.389923	0.26157	0.020106
ORB_manhattan	0.379763	0.397936	0.268128	0.028702
VGG_cosinus	0.1065	0.107851	0.047405	0.141562
VGG_euclidean	0.754663	0.788998	0.695522	0.103439
VGG_manhattan	0.75357	0.786984	0.692524	0.114153
NasNet_cosinus	0.118824	0.122606	0.079258	0.334193
NasNet_euclidean	0.70725	0.738717	0.638936	0.297163
NasNet_manhattan	0.700086	0.731934	0.634439	0.362533
EfficientNet_cosinus	0.179325	0.183471	0.083005	0.254814
EfficientNet_euclidean	0.743807	0.777928	0.68428	0.231353
EfficientNet_manhattan	0.739458	0.771273	0.673787	0.237542

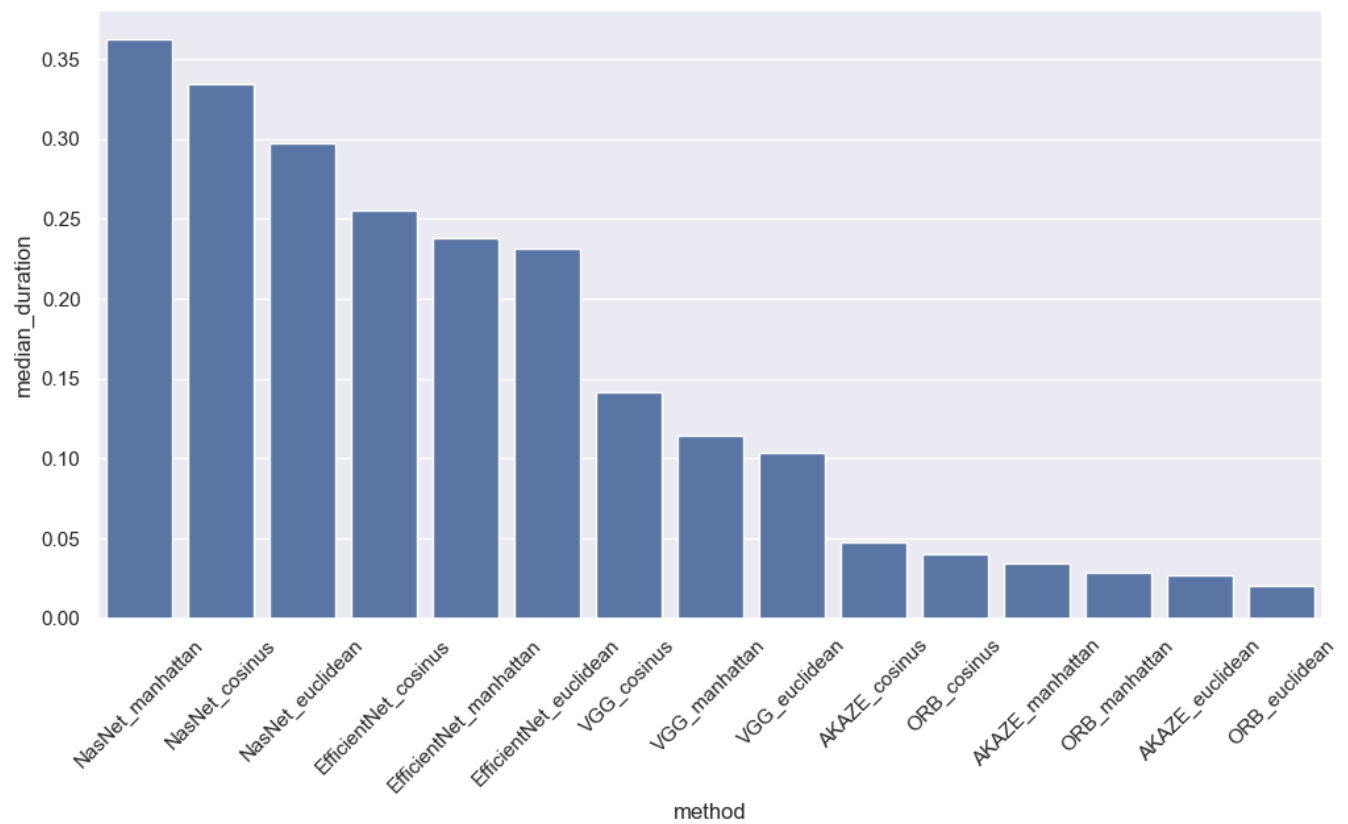
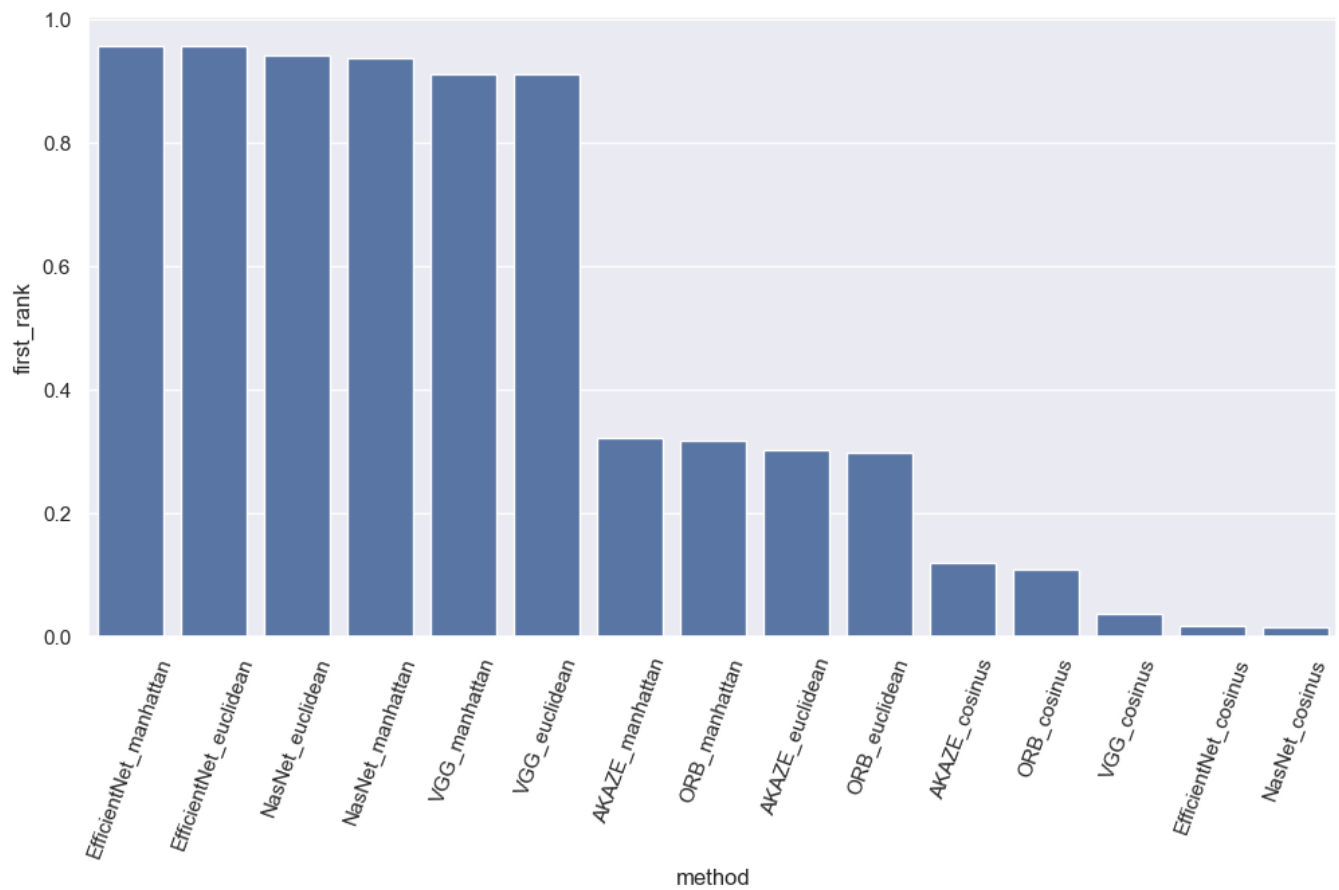




Styles

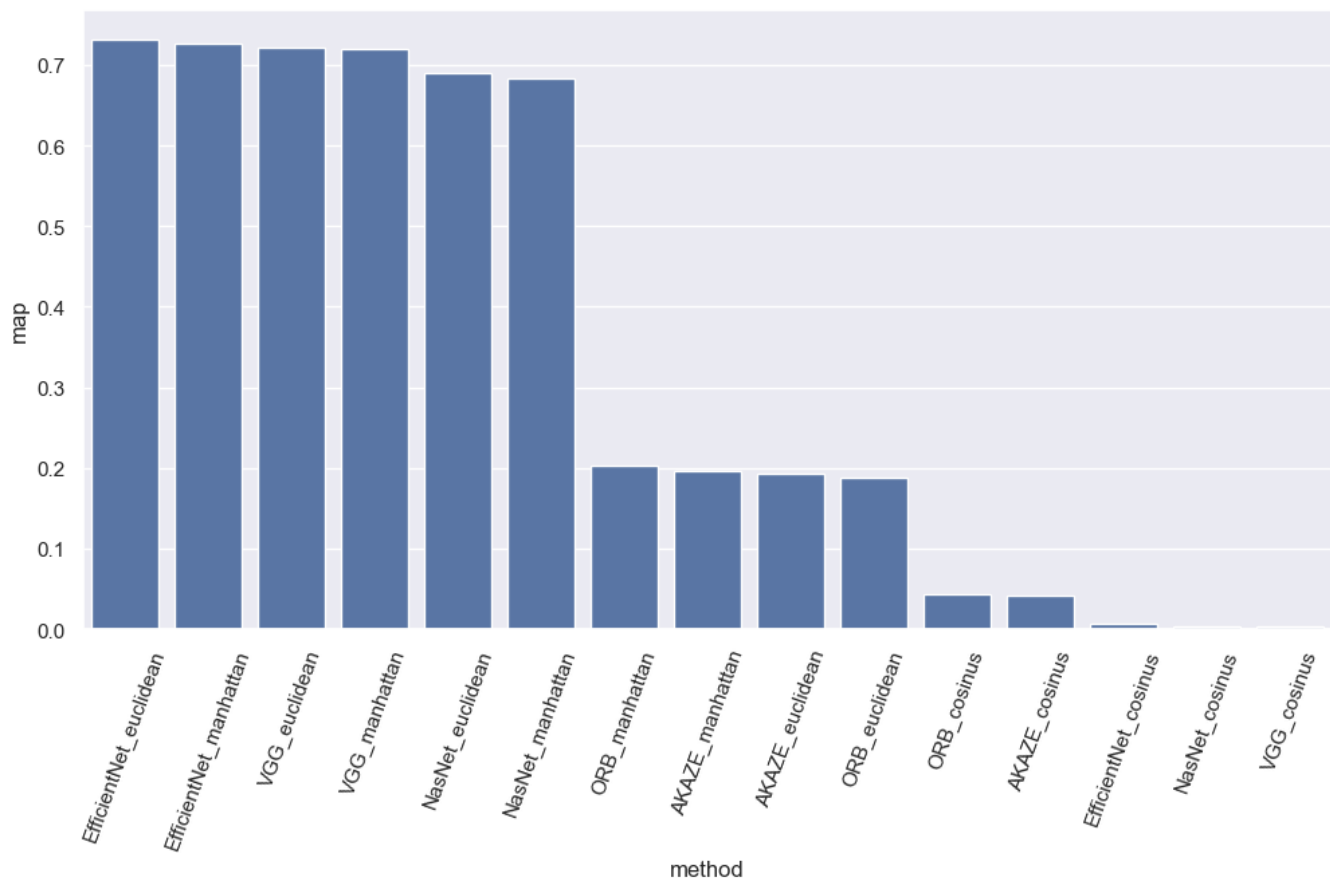
method	map	mrr	first_rank	median_duration
AKAZE_cosinus	0.199573	0.205184	0.118981	0.04769
AKAZE_euclidean	0.409305	0.431047	0.30148	0.027107
AKAZE_manhattan	0.426173	0.448473	0.320967	0.034335
ORB_cosinus	0.191999	0.195803	0.108863	0.040173
ORB_euclidean	0.409709	0.429314	0.297171	0.020106
ORB_manhattan	0.427056	0.446265	0.317407	0.028702
VGG_cosinus	0.045216	0.047024	0.037849	0.141562
VGG_euclidean	0.925721	0.938046	0.909687	0.103439
VGG_manhattan	0.925333	0.938205	0.909874	0.114153
NasNet_cosinus	0.027876	0.028774	0.015177	0.334193
NasNet_euclidean	0.951403	0.957279	0.941165	0.297163
NasNet_manhattan	0.946434	0.952114	0.936294	0.362533
EfficientNet_cosinus	0.054451	0.054397	0.018362	0.254814
EfficientNet_euclidean	0.961307	0.968056	0.955031	0.231353
EfficientNet_manhattan	0.961913	0.968765	0.955593	0.237542

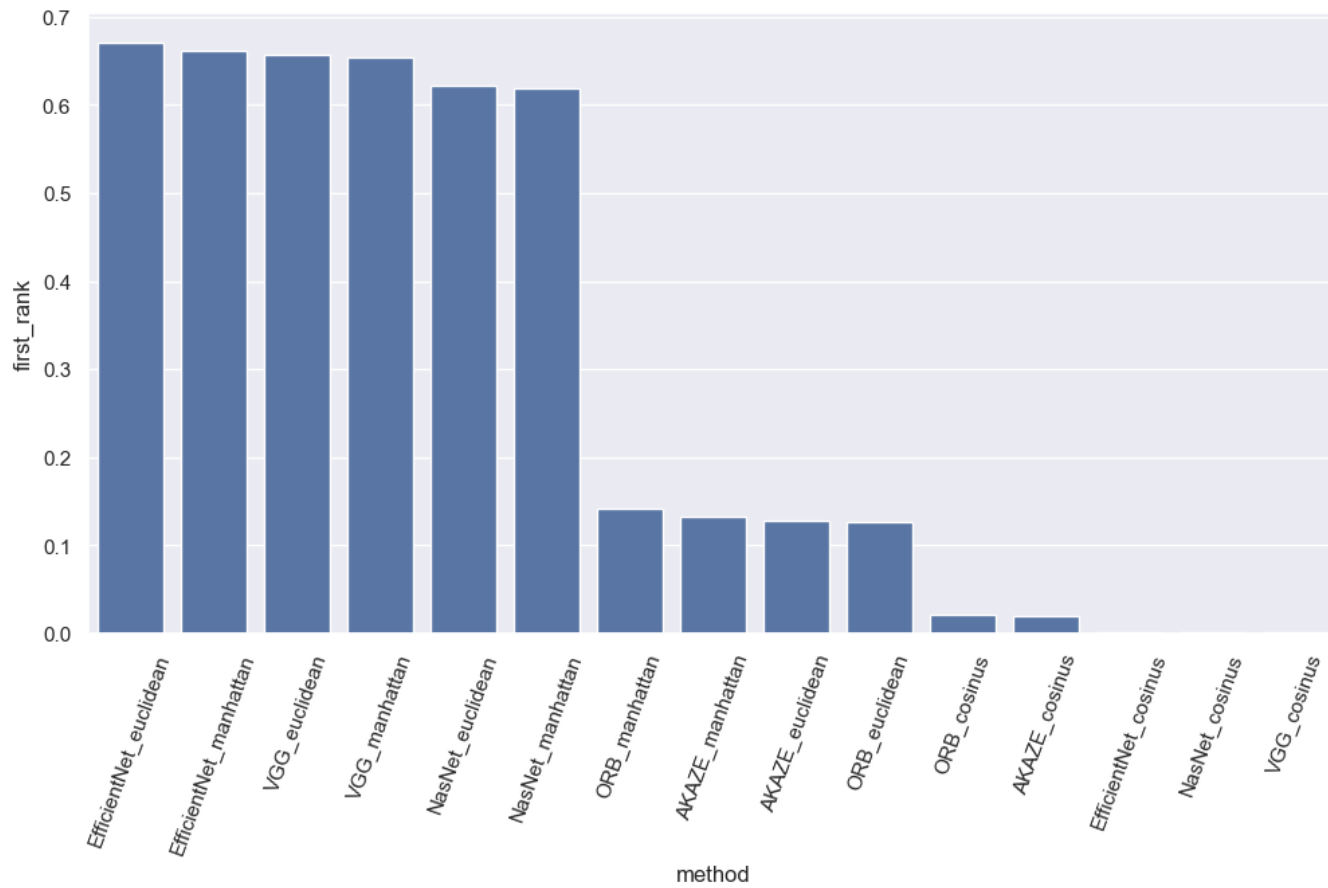
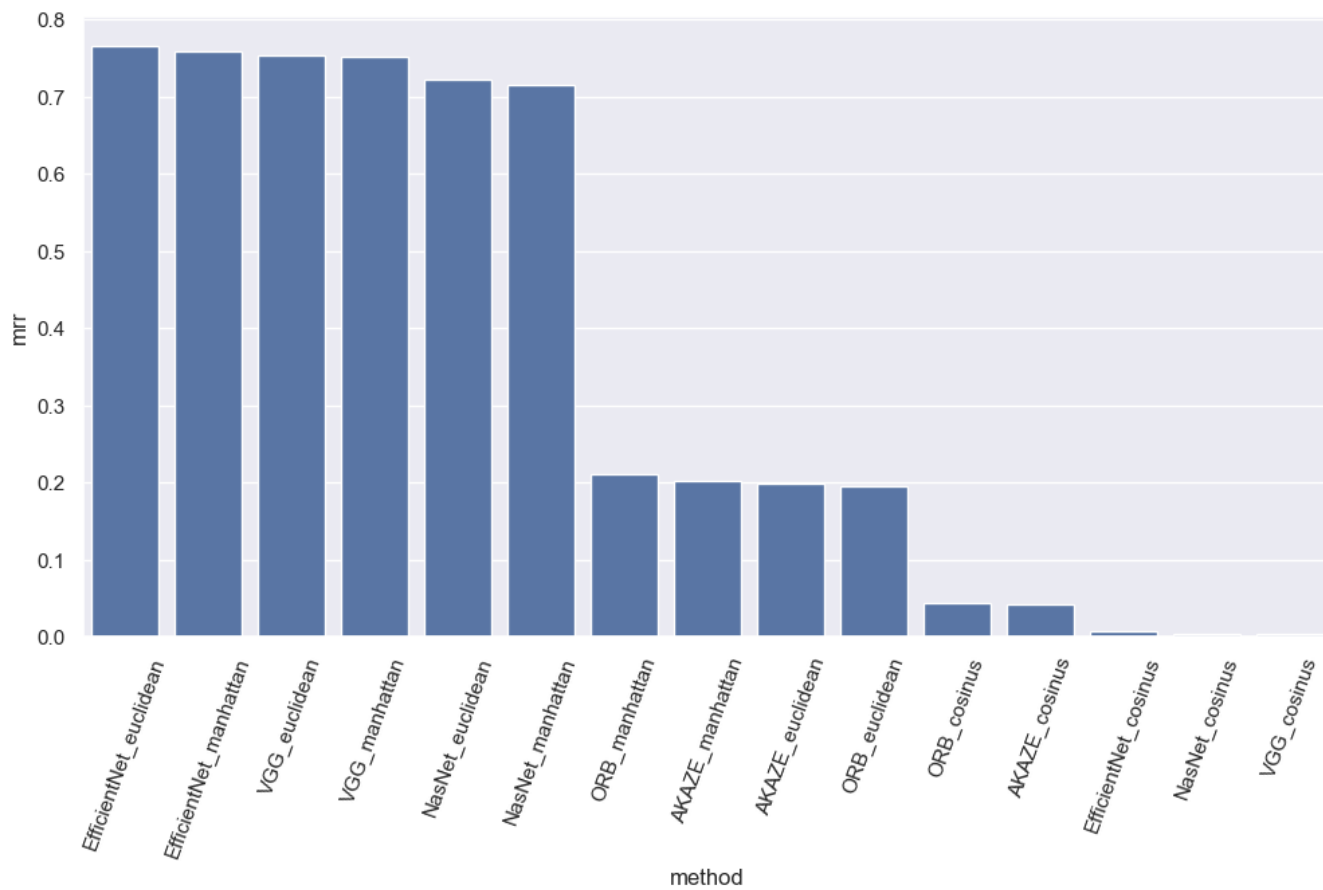


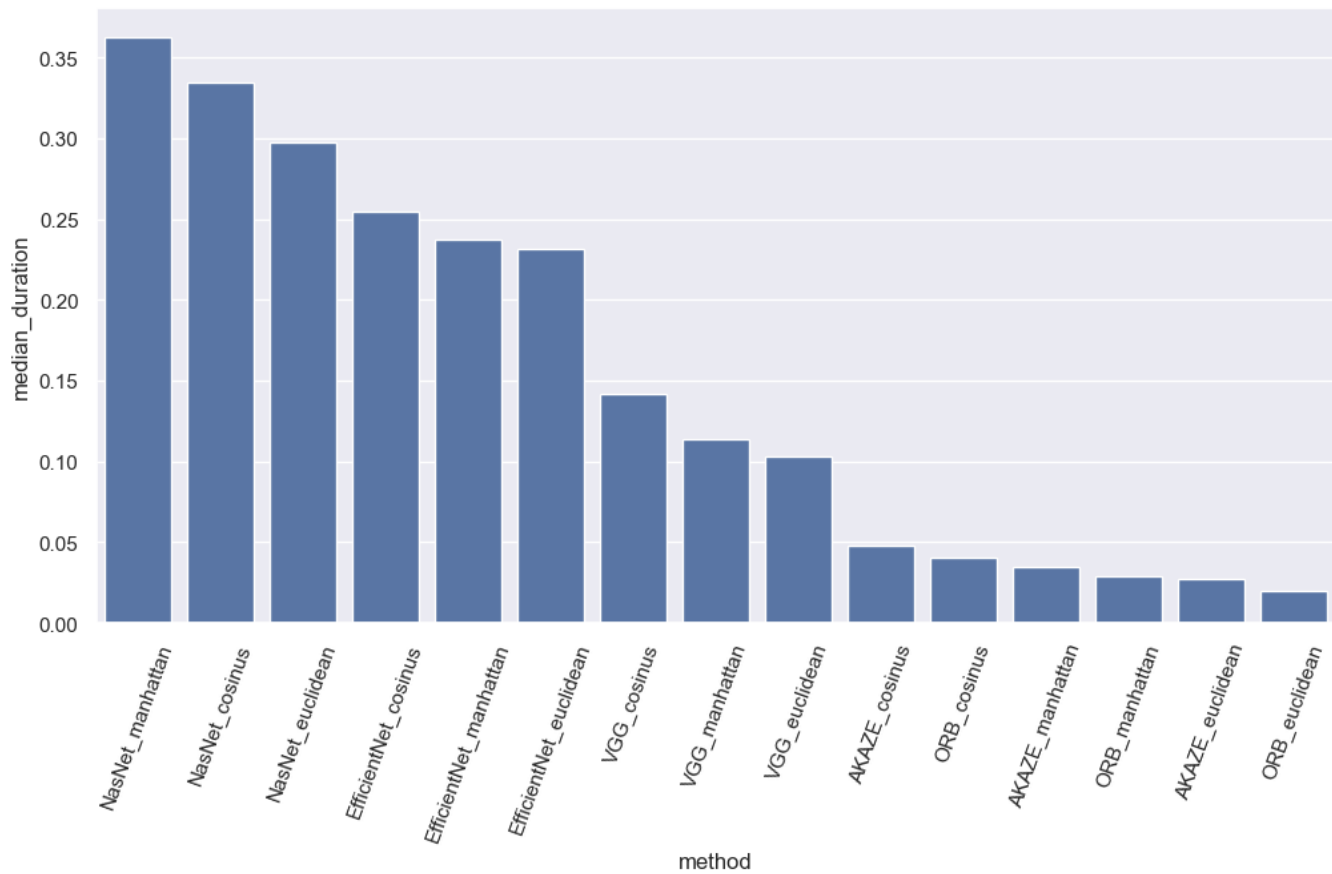


Colors & Styles

method	map	mrr	first_rank	median_duration
AKAZE_cosinus	0.042395	0.041974	0.020423	0.04769
AKAZE_euclidean	0.19283	0.198229	0.128349	0.027107
AKAZE_manhattan	0.196957	0.202442	0.133034	0.034335
ORB_cosinus	0.042541	0.042736	0.02136	0.040173
ORB_euclidean	0.188706	0.194775	0.126663	0.020106
ORB_manhattan	0.202788	0.210324	0.141653	0.028702
VGG_cosinus	0.002997	0.002982	0.001312	0.141562
VGG_euclidean	0.721211	0.753857	0.657298	0.103439
VGG_manhattan	0.719884	0.75178	0.6543	0.114153
NasNet_cosinus	0.003925	0.003925	0.001499	0.334193
NasNet_euclidean	0.689844	0.720961	0.621698	0.297163
NasNet_manhattan	0.683578	0.714996	0.619449	0.362533
EfficientNet_cosinus	0.006272	0.006314	0.001874	0.254814
EfficientNet_euclidean	0.731495	0.765105	0.670789	0.231353
EfficientNet_manhattan	0.726377	0.758535	0.661046	0.237542







Conclusion

Quelle que soit la méthode utilisée, la distance de cosinus présente la performance la moins satisfaisante. En revanche, les distances de Manhattan et euclidienne affichent des résultats assez similaires.

Les réseaux de neurones surpassent les descripteurs sur tous les critères d'évaluation.