



Big Data

Projet de reconnaissance d'images

-

Rapport 1ère partie

Membres du groupe : Jeffrey GONCALVES, Kévin LACOSTE

Enseignants : Claude BARRAS, Théophile SANCHEZ

Table des matières

Introduction	3
Méthodes	4
1. Nettoyage des données	4
2. Classifieur à distance minimum	5
3. Réduction de la dimension des vecteurs	6
4. Classifieur Scikit-Learn	6
Résultats	7
1. Classifieur à distance minimale (DMIN)	7
2. Support Vector Classifier (SVM)	16
3. KNeighborsClassifier (KNN)	21
Évaluation des résultats	28

Introduction

Ce rapport présente notre travail réalisé sur le projet de reconnaissance d'images, dont le but est de mettre en place et d'évaluer un système de reconnaissance d'images en Python.

Nous présenterons dans un premier temps nos réalisations en détail avec les méthodes que nous avons utilisées pour les implémenter, puis nous réaliserons une présentation de nos résultats, et enfin, nous évaluerons ces résultats et nous tirerons des conclusions sur l'efficacité des algorithmes que nous avons implémentés.

Méthodes

Cette section présente notre travail réalisé ainsi que les méthodes que nous avons utilisées pour implémenter chacun de nos algorithmes.

1. Nettoyage des données

Concernant le nettoyage des données, nous avons implémenté différents algorithmes de traitement d'images. Ces algorithmes s'appliquent à différents paramètres d'une image comme sa couleur (passage en niveaux de gris, binarisation), son contraste ou encore la création et modification d'histogramme (égalisation). Voici la liste des algorithmes implémentés :

- Passage en niveau de gris : Les images de base étant en couleur, nous avons eu l'idée de passer les images en niveaux de gris afin corriger certains défauts qui seraient présents sur l'image (notamment le flou). Pour passer d'une image en couleur à une image en niveaux de gris, nous appliquons la formule suivante à chaque composante RGB de chaque pixel de l'image :

$Y' = 0.299R + 0.587G + 0.114B$ où R est la composante rouge de l'image, G la composante verte et B la composante bleue.

- Binarisation : L'idée derrière cet algorithme est de fournir une valeur extrême (soit 0, soit 255) à chaque composante RGB de chaque pixel de l'image. Pour cela, on calcule tout d'abord une valeur moyenne de chaque composante puis on compare cette valeur à la valeur actuelle de la composante de chaque pixel. Si la valeur actuelle est supérieure à la valeur moyenne, la valeur de la composante sera fixée à 255 et 0 dans le cas contraire. Si on donne une image en couleur en paramètre, les pixels ne pourront donc avoir uniquement 8 couleurs différentes (blanc, noir, rouge, bleu, vert, cyan, magenta et jaune). De plus, si on donne une image en niveaux de gris en paramètre, les pixels ne pourront donc avoir que 2 valeurs différentes (soit blanc, soit noir).

- Égalisation d'histogramme : Nous avons eu aussi l'idée d'associer à chaque image un histogramme et réaliser une égalisation de chaque histogramme afin d'améliorer le contraste de chaque image. Pour égaliser nos histogrammes, nous avons utilisé les formules données par ce site (http://www.tsi.enst.fr/pages/enseignement/ressources/mti/egal-histo/rapport.htm#_To_c36969452), ce qui a nécessité l'implémentation d'un algorithme créant un histogramme normalisé pour une image donnée en paramètre ainsi que l'implémentation créant un histogramme cumulé d'une image.
- Augmentation du contraste : Cet algorithme utilise plusieurs formules que nous avons trouvé sur Internet. A cause de notre méconnaissance dans la mise en pratique de ces formules, cet algorithme n'est pas vraiment optimal et les résultats le concernant ne sont pas très satisfaisants, ce qui a conduit à l'implémentation de l'égalisation d'histogramme.
- Expansion dynamique : Le but cherché ici est de répartir les valeurs de pixels d'une image de manière plus uniforme sans pour autant chercher à l'uniformiser complètement. En effectuant ce pré-traitement, les valeurs de pixels très présentes dans l'image sont réparties plus équitablement. De même que pour l'égalisation d'histogramme, les formules utilisées viennent de http://www.tsi.enst.fr/pages/enseignement/ressources/mti/egal-histo/rapport.htm#_To_c36969452, et ont été implémentées sous la forme d'une fonction utilisant, tout comme l'égalisation d'histogramme, les fonctions d'histogramme et d'histogramme cumulé.

2. Classifieur à distance minimum

Concernant le classifieur à distance minimum, nous avons choisi d'implémenter une classe **DMIN** ayant comme attribut un dictionnaire **barycenters**, qui contient les barycentres des classes des données données en paramètre de la fonction **train** qui sépare les données selon leur classe (via leur label) pour ensuite calculer le barycentre de chaque classe, grâce à la fonction **numpy.mean**. Une fois que le barycentre de chaque classe a été calculé, nous stockons ces barycentres dans **barycenters**.

Après avoir lancé **train**, nous utilisons la fonction **find_class** (pour une image) ou **find_class_array** (pour un ensemble de d'images). Ces fonctions calculent la norme matricielle entre une image et les différents barycentres calculés dans **train** et va renvoyer l'identifiant de la classe (ou une liste d'identifiants dans le cas de **find_class_array**) correspondant à la distance la plus petite (et donc ainsi classer l'image ou l'ensemble d'images).

Enfin, nous comparons le label de chaque image avec son identifiant de classe associé dans la fonction **calculate_succes_rate**. À chaque fois que le label correspond

bien à l'identifiant, nous incrémentons le nombre de succès et nous renvoyons ce nombre ainsi que le pourcentage de succès par rapport au nombre de comparaisons réalisées.

3. Réduction de la dimension des vecteurs

L'ACP, pour Analyse en composante principale, a permis de réduire la dimension des vecteurs représentant les images, la faisant passer de 3072 (32 x 32 pixels, 3 valeurs (RGB) par pixel) à environ 100. Cette réduction est utile lors de l'utilisation d'algorithmes de classification ayant une complexité polynomiale élevée lors du traitement (ex : SVC, complexité sur la dimension du vecteur en n^3), cependant dans la quasi-totalité des cas, les performances du classifieur s'en trouvent réduites.

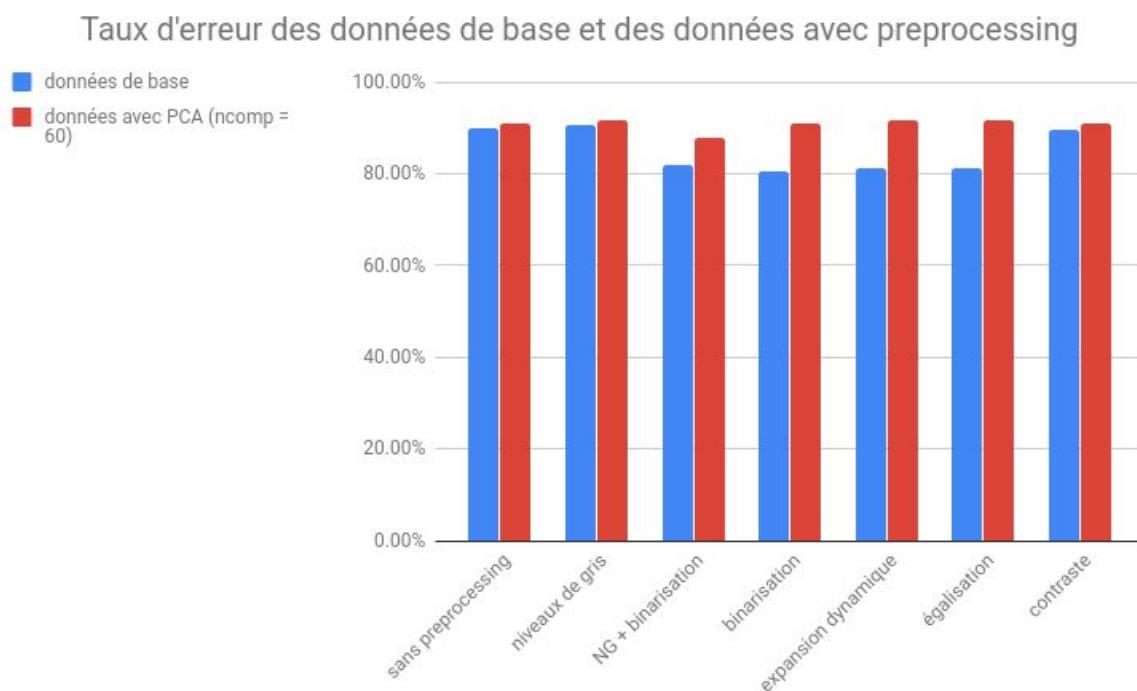
4. Classifieur Scikit-Learn

Nous avons implémenté parmi les classifieurs de scikit-learn le classifieur **SVC** (sklearn.svm.SVC) et le classifieur **KNeighborsClassifier** (sklearn.neighbors.KNeighborsClassifier). D'autres classifieurs existent dans Scikit-learn, comme par exemple **LinearSVC**, **NuSVC**, **NearestNeighbors**, etc. , mais nous avons décidé de ne pas les implémenter car les **SVC** et **KNeighborsClassifier** sont les plus rapides à implémenter, et, au final, ont été en général les plus performants, notamment le **SVC** avec le kernel réglé sur 'poly'

Résultats

1. Classifieur à distance minimale (DMIN)

Concernant le classifieur à distance minimum, nous avons choisi de séparer les résultats en deux catégories : une catégorie où aucune réduction de vecteur n'est appliqué et une autre catégorie où une réduction de vecteurs à 60 composantes est appliquée. On obtient donc la courbe des taux d'erreur suivante en fonction des différents type de preprocessing :



De plus, nous obtenons les matrices de confusion suivantes :

- **Données sans ACP appliquée :**

- Sans preprocessing :

Classes prédites	Vraies classes									
	0	1	2	3	4	5	6	7	8	9
0	597	36	200	31	22	41	17	613	28	159
1	1812	263	217	59	52	227	220	1519	100	630
2	1424	131	364	49	46	192	85	1267	31	560
3	959	115	187	46	21	134	106	843	33	438
4	781	72	236	29	56	119	96	755	35	344
5	826	56	212	52	16	109	39	741	25	308
6	695	55	203	49	16	62	30	587	23	257
7	785	49	96	21	17	109	47	684	16	195
8	546	32	194	36	23	57	18	516	29	209
9	540	38	161	25	10	79	45	500	19	178

- Niveaux de gris :

Classes prédites	Vraies classes									
	0	1	2	3	4	5	6	7	8	9
0	733	26	62	16	10	57	31	444	43	322
1	1868	111	318	42	59	80	152	983	122	1364
2	1658	155	197	56	50	78	77	972	38	868
3	1090	92	115	43	25	78	55	633	68	683
4	836	68	116	23	130	44	185	477	29	615
5	901	52	77	18	13	161	72	551	59	480
6	759	52	60	21	17	87	150	461	16	354
7	769	66	181	15	13	34	29	540	8	364
8	616	46	61	13	16	112	99	335	32	330
9	641	52	73	14	9	58	34	376	30	308

○ Binarisation :

Classes prédites	Vraies classes									
	0	1	2	3	4	5	6	7	8	9
0	516	424	107	68	140	102	109	162	50	66
1	1398	599	263	190	928	412	204	891	96	118
2	320	1249	562	162	520	233	241	404	318	140
3	345	773	87	314	312	308	161	313	179	90
4	370	430	230	53	983	109	170	133	15	30
5	213	592	137	81	229	643	211	177	54	47
6	175	428	149	71	282	213	502	105	24	28
7	271	367	213	81	97	90	155	644	59	42
8	151	471	71	41	185	220	300	85	115	21
9	215	458	53	32	148	184	113	136	70	186

○ Niveaux de gris et binarisation :

Classes prédites	Vraies classes									
	0	1	2	3	4	5	6	7	8	9
0	490	492	90	77	142	96	116	143	39	490
1	1446	472	157	212	944	473	219	900	62	1446
2	298	1142	556	173	590	255	252	300	286	298
3	281	683	77	297	341	339	186	273	162	281
4	392	433	169	84	968	137	170	107	6	392
5	174	616	87	96	228	690	189	189	38	174
6	188	456	110	83	284	239	466	466	23	188
7	208	402	274	86	99	117	167	167	56	208
8	126	497	55	51	187	237	291	291	100	126
9	205	469	46	54	142	224	108	108	52	205

○ Egalisation :

Classes prédites	Vraies classes									
	0	1	2	3	4	5	6	7	8	9
0	508	484	128	53	120	98	77	167	56	56
1	1416	604	213	283	749	447	230	924	118	118
2	365	1119	536	198	501	228	222	359	384	237
3	308	695	109	323	306	296	146	329	207	163
4	338	453	213	62	921	141	162	143	23	67
5	174	572	147	80	201	671	170	216	52	101
6	194	406	175	66	250	254	438	126	30	38
7	193	450	205	78	100	91	135	598	67	102
8	107	494	85	35	154	280	229	118	135	23
9	188	468	62	45	119	187	90	172	79	185

○ Augmentation du contraste :

Classes prédites	Vraies classes									
	0	1	2	3	4	5	6	7	8	9
0	695	44	77	32	31	69	33	118	34	611
1	1511	208	552	54	95	101	102	329	196	1951
2	1391	297	344	52	64	89	62	207	110	1533
3	927	195	188	51	37	86	42	155	130	1071
4	689	115	142	45	244	75	151	101	74	887
5	767	105	110	25	22	189	58	144	135	829
6	653	77	101	37	35	119	152	134	42	627
7	611	93	312	7	20	44	24	119	24	765
8	510	91	71	16	25	117	84	97	119	530
9	573	107	91	16	24	44	35	86	55	564

- Expansion dynamique :

Classes prédites	Vraies classes									
	0	1	2	3	4	5	6	7	8	9
0	508	484	128	53	120	98	77	167	56	56
1	1416	604	213	283	749	447	230	924	118	118
2	365	1119	536	198	501	228	222	359	384	237
3	308	695	109	323	306	296	146	329	207	163
4	338	453	213	62	921	141	162	143	23	67
5	174	572	147	80	201	671	170	216	52	101
6	194	406	175	66	250	254	438	126	30	38
7	193	450	205	78	100	91	135	598	67	102
8	107	494	85	35	154	280	229	118	135	23
9	188	468	62	45	119	187	90	172	79	185

- **Données avec ACP appliquée :**

- Sans preprocessing :

Classes prédites	Vraies classes									
	0	1	2	3	4	5	6	7	8	9
0	597	36	200	31	22	41	17	613	28	159
1	1812	263	217	59	52	227	220	1519	100	630
2	1424	131	364	49	46	192	85	1267	31	560
3	959	115	187	46	21	134	106	843	33	438
4	781	72	236	29	56	119	96	755	35	344
5	826	56	212	52	16	109	39	741	25	308
6	695	55	203	49	16	62	30	587	23	257
7	785	49	96	21	17	109	47	684	16	195
8	546	32	194	36	23	57	18	516	29	209
9	540	38	161	25	10	79	45	500	19	178

○ Niveaux de gris :

Classes prédites	Vraies classes									
	0	1	2	3	4	5	6	7	8	9
0	674	32	78	63	19	24	27	543	30	254
1	1807	199	329	41	38	182	203	1032	36	1232
2	1523	111	228	43	49	198	86	1160	47	704
3	1003	98	176	31	24	104	64	777	34	571
4	818	54	235	61	34	78	87	604	41	532
5	901	45	122	42	24	70	40	722	32	386
6	765	36	99	61	17	49	42	555	32	321
7	807	40	70	31	28	64	43	580	16	340
8	604	37	120	31	17	43	32	458	25	293
9	607	26	70	36	14	66	46	448	29	253

○ Binarisation :

Classes prédites	Vraies classes									
	0	1	2	3	4	5	6	7	8	9
0	215	422	66	149	234	109	199	263	97	70
1	726	776	476	156	838	737	336	652	152	196
2	436	1136	208	110	566	612	471	369	110	85
3	341	735	147	106	334	471	303	295	74	72
4	296	566	138	112	334	400	128	357	93	105
5	283	564	87	139	382	319	250	226	61	100
6	246	445	82	111	305	305	119	209	65	62
7	275	414	169	116	367	120	228	199	75	61
8	170	470	55	72	265	220	142	123	61	38
9	178	436	56	149	285	140	108	185	87	48

○ Niveaux de gris et binarisation :

Classes prédites	Vraies classes									
	0	1	2	3	4	5	6	7	8	9
0	329	182	137	49	467	58	54	271	40	157
1	481	1676	228	251	290	939	529	286	315	104
2	312	1413	118	96	489	876	499	174	64	108
3	255	871	138	84	270	556	405	147	62	94
4	285	701	128	91	371	361	234	165	45	142
5	325	572	131	46	419	294	159	259	33	146
6	290	348	125	75	421	162	76	196	24	160
7	173	626	75	65	280	296	270	97	78	59
8	249	367	70	48	354	151	90	232	15	84
9	259	340	83	33	350	109	57	258	23	83

○ Egalisation :

Classes prédites	Vraies classes									
	0	1	2	3	4	5	6	7	8	9
0	92	450	128	108	130	279	142	197	94	124
1	1232	783	265	192	801	466	431	601	146	182
2	1010	645	137	192	861	417	236	380	87	184
3	453	539	147	152	372	262	297	299	94	267
4	98	586	209	117	289	288	269	249	98	320
5	128	567	197	103	278	262	258	297	156	138
6	82	425	123	129	212	211	218	386	94	97
7	484	411	87	133	372	216	101	102	34	79
8	87	404	70	77	162	259	160	232	101	108
9	69	366	92	108	198	309	121	193	77	62

○ Augmentation du contraste :

Classes prédites	Vraies classes									
	0	1	2	3	4	5	6	7	8	9
0	521	79	196	41	63	54	57	79	27	611
1	1735	217	291	34	82	218	128	262	64	1951
2	1415	144	248	41	87	181	85	254	62	1533
3	880	88	246	36	59	122	82	166	44	1071
4	685	126	329	23	60	73	61	128	52	887
5	690	101	228	27	54	106	52	147	70	829
6	567	84	247	43	42	65	50	113	39	627
7	779	86	57	12	27	76	43	98	34	765
8	468	80	195	30	43	65	49	90	41	530
9	440	65	171	33	46	71	38	73	69	564

○ Expansion dynamique :

Classes prédites	Vraies classes									
	0	1	2	3	4	5	6	7	8	9
0	92	449	129	107	130	280	141	195	95	126
1	1232	784	268	192	799	464	434	601	145	180
2	1013	645	138	190	858	420	235	380	86	184
3	455	538	147	154	372	261	299	299	95	262
4	98	585	212	115	292	284	267	251	99	320
5	129	568	197	104	279	265	254	295	156	137
6	82	426	123	130	211	212	218	385	93	97
7	485	413	87	132	370	217	101	101	35	78
8	87	404	70	77	161	258	160	232	102	109
9	69	364	92	108	196	310	120	197	77	62

Les tableau des taux d'erreur et des temps d'exécution sont les suivants (certains temps d'exécution apparaissent en rouge et correspondent à un temps d'exécution sur une machine avec un processeur Intel i7 qui est beaucoup plus rapide que la machine qui a réalisé les autres tests et qui utilise quant à elle une processeur Intel Celeron) :

- **Données sans ACP appliquée :**

	sans préprocessing	Niveaux de gris	NG + binarisation	binarisation	expansion dynamique	égalisation	contraste
Taux d'erreur	89.91%	90.76%	81.88%	80.55%	81.10%	81.10%	89.69%
Temps d'exécution	18.06 s	20.00 s	19.60 s	19.67 s	19.35 s	19.45 s	4.36 s

- **Données avec ACP appliquée :**

	sans préprocessing	Niveaux de gris	NG + binarisation	binarisation	expansion dynamique	égalisation	contraste
Taux d'erreur	90.95%	91.79%	87.93%	90.82%	91.52%	91.56%	90.95%
Temps d'exécution	32.35s	4.45 min	3.58 min	32.78 s	5.24 min	5.41 min	32.35 s

2. Support Vector Classifier (SVM)

Chaque test a été effectué avec 10 000 images d'entraînement et le set complet des images de test sans ACP. Cependant, pour une raison inconnue, le SVC avec ACP met trop de temps à s'exécuter et ne sera pas affiché ici.

En addition à cela, il est important de noter que les résultats suivant ont été trouvés avec un processeur i7 (notamment en termes de temps de calcul, en comparaison aux parties précédentes où un processeur Celeron a été utilisé).

Performances du classifieur :

	Sans ACP (10 000 images d'entraînement)
Sans pré-processing	60.71 %
Noir et blanc	63.94 %
Binarisation (Noir et Blanc)	67.79 %
Binarisation (Couleur)	66.11 %
Contraste à 95	44.98 %
Expansion dynamique	67.83 %
Egalisation d'histogramme	67.91 %

Temps de calcul du classifieur :

	Sans ACP (10 000 images d'entraînement)
Sans pré-processing	15 min 59 sec
Noir et blanc	15 min 49 sec
Binarisation (Noir et Blanc)	21 min 58 sec
Binarisation (Couleur)	25 min 19 sec
Contraste à 95	15 min 38 sec
Expansion dynamique	16 min 51 sec
Egalisation d'histogramme	17 min 01 sec

Matrices de confusion (sans ACP) :

- Sans pré-processing :

Classes prédites	Vraies classes									
	0	1	2	3	4	5	6	7	8	9
0	1039	82	72	86	57	76	120	47	66	99
1	91	3797	229	246	239	137	87	132	71	70
2	90	298	2601	333	163	123	77	186	129	149
3	80	308	174	1487	84	206	51	80	158	254
4	69	168	80	60	1791	49	107	27	86	86
5	72	143	83	237	90	1259	237	38	138	87
6	107	99	65	59	206	139	1137	28	97	40
7	46	278	287	76	37	41	49	1097	34	74
8	99	82	79	69	103	80	299	25	730	94
9	182	88	77	51	75	79	63	31	82	867

- Noir et blanc :

Classes prédites	Vraies classes									
	0	1	2	3	4	5	6	7	8	9
0	1085	60	66	94	46	52	135	40	58	108
1	69	3948	188	213	188	110	98	140	68	77
2	69	249	2792	308	131	98	76	158	108	160
3	67	245	174	1583	83	173	56	94	127	280
4	59	165	73	54	1833	40	111	26	53	109
5	51	123	71	248	92	1315	217	48	137	82
6	95	84	46	64	164	113	1253	22	100	36
7	46	245	270	84	26	38	35	1147	42	86
8	102	67	57	79	101	69	283	27	776	99
9	182	67	65	58	61	65	69	28	87	913

- Binarisation (Noir et Blanc) :

Classes prédites	Vraies classes									
	0	1	2	3	4	5	6	7	8	9
0	1170	75	64	70	66	57	90	38	35	79
1	32	4267	173	152	158	56	52	103	52	54
2	43	367	3022	230	87	67	39	192	80	72
3	35	322	254	1611	67	185	42	71	90	205
4	42	206	72	44	1918	45	94	26	22	54
5	48	206	85	208	65	1401	152	25	87	107
6	110	116	56	51	216	154	1136	16	95	27
7	25	265	233	61	19	17	19	1330	14	36
8	108	131	92	83	92	61	183	20	812	78
9	139	134	86	40	45	52	29	30	60	980

- Binarisation (Couleur) :

Classes prédites	Vraies classes									
	0	1	2	3	4	5	6	7	8	9
0	1126	104	76	75	72	58	90	24	46	73
1	47	4288	185	173	141	47	52	82	38	46
2	56	456	2898	247	114	69	33	113	84	79
3	46	402	242	1584	75	167	34	68	90	174
4	40	218	77	51	1883	31	113	23	32	55
5	42	218	82	169	91	1380	173	28	86	115
6	94	140	61	48	227	148	1217	10	40	23
7	23	317	264	68	26	28	26	1217	10	40
8	109	141	96	84	118	67	177	18	765	85
9	138	127	110	39	50	52	42	24	47	966

- Contraste à 95 :

Classes prédites	Vraies classes									
	0	1	2	3	4	5	6	7	8	9
0	826	74	65	70	53	50	398	31	77	100
1	161	3020	175	302	121	129	491	101	318	281
2	271	318	1895	335	105	132	340	123	372	258
3	152	332	167	1037	51	212	295	69	210	357
4	95	219	77	109	1150	50	450	10	245	118
5	123	138	85	271	51	775	560	23	217	141
6	150	89	54	59	94	96	1223	12	166	34
7	91	246	298	138	26	89	178	618	133	202
8	132	85	42	81	63	68	577	10	553	49
9	178	100	70	82	48	59	236	18	191	613

- Expansion dynamique :

Classes prédites	Vraies classes									
	0	1	2	3	4	5	6	7	8	9
0	1229	68	59	77	35	50	79	20	42	85
1	35	4122	204	217	193	52	62	102	56	56
2	41	331	2984	303	68	54	50	130	102	86
3	42	240	221	1719	67	168	34	72	109	210
4	41	188	87	76	1839	47	111	27	41	66
5	41	133	87	273	64	1414	146	35	90	101
6	85	88	70	59	169	139	1213	30	88	36
7	24	276	290	62	17	40	18	1217	26	49
8	66	97	61	94	73	95	203	20	873	88
9	105	77	73	66	33	66	34	32	62	1047

- Egalisation d'histogramme :

Classes prédites	Vraies classes									
	0	1	2	3	4	5	6	7	8	9
0	1229	68	59	77	35	50	79	20	42	85
1	35	4122	204	217	193	52	62	102	56	56
2	41	331	2984	303	68	54	50	130	102	86
3	42	240	221	1719	67	168	34	72	109	210
4	41	188	87	76	1839	47	111	27	41	66
5	41	133	87	273	64	1414	146	35	90	101
6	85	88	70	59	169	139	1213	30	88	36
7	24	276	290	62	17	40	18	1217	26	49
8	66	97	61	94	73	95	203	20	873	88
9	105	77	73	66	33	66	34	32	62	1047

3. KNeighborsClassifier (KNN)

Chaque test a été effectué avec 10 000 images d'entraînement et le set complet des images de test sans ACP, et sur le set complet des images d'entraînement (73 257) et le set complet des images de test avec l'ACP (nombre de composantes : 60).

Performances du classifieur :

	Sans ACP (10 000 images d'entraînement)	Avec ACP (nombre de composantes = 60)
Sans pré-processing	36.24 %	15.11 %
Noir et blanc	39.61 %	17.44 %
Binarisation (Noir et Blanc)	49.88 %	10.63 %
Binarisation (Couleur)	45.89 %	14.31 %
Contraste à 95	37.42 %	15.37 %
Expansion dynamique	50.89 %	17.51 %
Egalisation d'histogramme	50.90 %	17.53 %

Temps de calcul du classifieur :

	Sans ACP (10 000 images d'entraînement)	Avec ACP (nombre de composantes = 60)
Sans pré-processing	18 min 53 sec	2 min 3 sec
Noir et blanc	18 min 9 sec	1 min 59 sec
Binarisation (Noir et Blanc)	22 min 13 sec	4 min 19 sec
Binarisation (Couleur)	22 min 15 sec	4 min 29 sec
Contraste à 95	20 min 14 sec	2 min 14 sec
Expansion dynamique	22 min 32 sec	4 min 8 sec
Egalisation d'histogramme	22 min 25 sec	4 min 10 sec

Matrices de confusion (sans ACP, 10000 images d'entraînement) :

- Sans pré-processing :

Classes prédites	Vraies classes										
		0	1	2	3	4	5	6	7	8	9
	0	457	348	209	188	134	76	117	31	63	121
	1	48	3809	407	284	266	80	39	72	42	52
	2	76	1272	1727	383	193	87	75	133	82	121
	3	54	1058	472	722	144	140	53	59	74	106
	4	53	787	199	158	1056	55	79	23	57	56
	5	75	680	282	381	151	379	172	32	118	114
	6	107	469	183	158	257	150	414	16	142	81
	7	27	775	433	182	68	46	22	394	21	51
	8	95	358	224	192	161	102	211	28	185	104
	9	166	394	216	166	105	84	38	31	74	291

- Noir et blanc :

Classes prédites	Vraies classes										
		0	1	2	3	4	5	6	7	8	9
	0	502	314	182	172	119	93	136	30	74	122
	1	46	4017	277	252	227	63	42	74	39	62
	2	66	1244	1877	350	155	88	57	133	80	99
	3	51	1025	423	800	147	147	50	50	93	96
	4	45	727	163	136	1225	36	71	14	49	57
	5	67	675	239	389	144	441	175	28	116	110
	6	120	417	155	141	303	165	430	12	159	75
	7	30	756	409	171	58	40	25	466	19	45
	8	98	340	185	192	138	115	221	21	242	108
	9	143	339	230	181	106	86	83	27	90	310

- Binairisation (Noir et Blanc) :

Classes prédites	Vraies classes									
	0	1	2	3	4	5	6	7	8	9
0	855	225	70	89	114	60	138	16	73	104
1	48	4485	137	122	117	29	48	58	28	27
2	62	1170	2077	304	106	45	69	135	108	73
3	65	984	328	1041	73	117	47	60	91	76
4	68	595	74	53	1582	28	66	12	25	20
5	89	630	155	289	114	641	258	18	111	79
6	128	338	102	99	344	153	619	13	128	53
7	29	715	236	83	29	21	19	851	12	24
8	139	312	136	136	127	94	270	11	365	70
9	211	330	142	89	72	61	99	33	88	470

- Binarisation (Couleur) :

Classes prédites	Vraies classes									
	0	1	2	3	4	5	6	7	8	9
0	793	282	87	84	115	60	123	23	72	105
1	44	4383	163	164	131	39	46	60	37	32
2	72	1274	1896	345	130	59	72	114	113	74
3	63	1069	290	937	93	125	60	74	96	75
4	71	725	99	67	1360	30	85	22	33	31
5	91	684	167	316	126	571	189	29	125	86
6	126	420	124	132	296	166	537	12	118	46
7	27	798	227	91	28	31	25	742	21	29
8	133	366	155	160	117	98	229	11	328	63
9	189	395	171	106	76	61	80	34	84	399

- Contraste à 95 :

Classes prédites	Vraies classes									
	0	1	2	3	4	5	6	7	8	9
0	478	352	217	166	128	73	115	31	73	111
1	45	3792	374	310	263	86	43	91	39	56
2	66	1239	1780	385	235	86	60	133	68	97
3	44	1039	436	778	187	134	46	66	53	99
4	61	768	193	134	1138	35	70	27	44	53
5	60	682	285	388	173	404	149	35	106	102
6	109	487	196	148	264	128	442	19	122	62
7	35	757	417	164	72	41	24	439	23	47
8	97	383	222	189	163	97	209	25	174	101
9	150	413	237	161	112	67	61	34	43	317

- Expansion dynamique :

Classes prédites	Vraies classes									
	0	1	2	3	4	5	6	7	8	9
0	892	188	95	82	121	50	135	15	69	97
1	54	4304	208	162	158	31	44	64	36	38
2	58	1008	2152	404	127	62	59	117	82	80
3	63	821	318	1156	99	122	46	58	100	99
4	65	522	77	98	1556	31	77	15	45	37
5	77	490	117	351	141	730	230	25	114	109
6	140	285	77	115	265	186	719	17	131	42
7	40	603	337	106	24	31	23	800	19	36
8	126	248	108	154	140	110	282	13	401	78
9	205	281	146	106	75	68	76	34	66	538

- Egalisation d'histogramme : similaire à l'expansion dynamique.

Matrices de confusion (avec ACP à 60 composantes) :

- Sans pré-processing :

Classes prédites	Vraies classes									
	0	1	2	3	4	5	6	7	8	9
0	120	763	221	189	99	109	72	61	45	65
1	124	1861	884	536	544	274	328	76	265	180
2	146	1587	764	422	216	246	166	226	203	173
3	100	1283	405	249	200	140	164	88	147	106
4	77	812	289	258	587	131	140	60	100	69
5	110	996	327	254	166	144	128	60	83	116
6	76	830	271	297	162	86	61	71	51	72
7	72	718	338	218	133	186	148	43	115	48
8	67	714	211	182	135	101	86	57	58	49
9	75	673	202	180	121	106	98	27	64	49

- Noir et blanc :

Classes prédites	Vraies classes									
	0	1	2	3	4	5	6	7	8	9
0	71	787	265	227	107	73	77	48	55	34
1	127	2388	608	365	556	223	304	94	235	199
2	170	1359	806	420	272	250	219	186	234	233
3	119	1142	510	310	180	134	163	91	109	124
4	43	1072	227	256	580	116	72	43	52	62
5	114	909	362	318	195	123	105	72	97	89
6	103	760	254	303	183	85	79	43	78	89
7	71	760	267	201	148	227	158	31	111	45
8	92	612	218	262	136	79	76	45	77	63
9	72	595	171	189	100	172	104	24	93	75

- Binairisation (Noir et Blanc) :

Classes prédites	Vraies classes									
	0	1	2	3	4	5	6	7	8	9
0	48	647	252	157	358	65	50	79	45	43
1	284	1284	918	589	262	459	440	135	453	275
2	135	981	477	455	625	348	464	79	437	148
3	116	736	393	413	262	309	181	55	294	123
4	96	785	436	417	113	205	108	122	117	124
5	91	901	390	246	161	156	70	51	119	199
6	51	865	285	239	166	91	43	70	57	110
7	44	700	311	172	197	121	147	40	220	67
8	46	596	213	250	201	89	56	54	87	68
9	57	647	248	134	164	61	58	57	64	105

- Binarisation (Couleur) :

Classes prédites	Vraies classes									
	0	1	2	3	4	5	6	7	8	9
0	33	660	306	264	72	65	71	71	82	120
1	159	1987	645	470	730	278	236	329	191	74
2	234	1192	490	390	576	216	289	132	433	197
3	75	883	331	360	254	312	246	124	194	103
4	61	890	398	286	130	188	87	183	88	212
5	47	651	294	397	210	187	192	52	179	75
6	40	589	236	400	123	191	113	46	139	100
7	71	735	332	168	182	90	107	98	127	109
8	48	503	196	302	133	100	115	36	139	88
9	32	564	304	190	116	65	91	55	90	88

- Contraste à 95 :

Classes prédites	Vraies classes									
	0	1	2	3	4	5	6	7	8	9
0	42	738	262	199	167	62	108	70	59	37
1	236	2112	566	708	366	268	282	76	263	222
2	273	1515	810	381	285	231	179	127	196	152
3	105	1226	444	313	213	139	134	76	132	100
4	58	1292	313	229	296	110	60	50	60	55
5	74	997	357	266	184	160	134	43	105	64
6	56	739	360	215	202	126	87	41	100	51
7	93	906	331	173	102	125	84	75	78	52
8	47	744	233	184	133	110	76	25	66	42
9	59	733	222	161	116	88	66	44	67	39

- Expansion dynamique :

Classes prédites	Vraies classes									
	0	1	2	3	4	5	6	7	8	9
0	80	738	183	189	173	81	72	50	82	96
1	176	2706	406	352	329	202	294	189	297	148
2	171	1382	658	427	550	310	210	131	179	131
3	113	1045	434	378	196	165	151	115	181	104
4	71	753	361	367	256	142	136	231	137	69
5	52	780	412	311	186	142	128	114	159	100
6	69	827	201	212	208	91	93	78	121	66
7	112	531	221	257	360	109	86	63	103	177
8	67	702	182	191	126	72	103	40	113	64
9	55	587	225	207	130	106	66	55	94	70

- Egalisation d'histogramme : similaire à l'expansion dynamique.

Évaluation des résultats

En réunissant tous les résultats évoqués précédemment, il est possible de s'apercevoir de plusieurs choses :

- L'ACP a une fâcheuse tendance à concentrer les résultats de prédiction sur 1 ou 2 labels, ce qui au final ruine les performances de nos classifieurs. Il est probable que nous en faisons une mauvaise utilisation, ce genre d'algorithme étant en général utilisé pour supprimer le bruit d'une image en appliquant l'ACP puis en inversant la procédure.
→ `pca_data = pca.fit_transform(X)` suivi d'un `X = pca.inverse_transform(pca_data)`
- Le DMIN s'est révélé peu efficace, allant au maximum à 20 % de précision avec le meilleur pré-traitement possible. Cela est sûrement dû à l'utilisation de normes naïves (normes de Frobenius) peu adaptées à nos données. Il est hautement probable qu'en modifiant la norme utilisée pour le classifieur par distance, les performances seraient meilleures.
- Le meilleur de nos classifieurs est le **SVC**, mais pour obtenir ces résultats, il a fallu modifier les paramètres du classifieur : sans la modification du kernel sur le réglage 'poly', ce classifieur restait bloqué à 30 % de réussite sur les fichiers test. Cela met en avant les défauts du SVC :
 - Le temps d'entraînement du SVC peut rapidement exploser à cause de sa complexité en n^3 . De ce fait, quand un entraînement sur 10 000 images met en moyenne 20 minutes sur un processeur i7, le même entraînement sur 73000 images peut mettre un temps irraisonnable (il nous est arrivé de le faire tourner seul sur 24h sans que l'entraînement ne soit fini).
 - Les paramètres de réglages du SVC sont cruciaux, et s'ils ne sont pas bien définis, celui-ci ne présente que des performances médiocres. Il s'avère donc qu'il est difficile de correctement le régler, surtout avec le temps d'entraînement qui peut "exploser" rapidement.
- Le **KNeighborsClassifier** reste assez performant malgré ses capacités de classification inférieures au SVC. Cependant, ce classifieur est plus simple et plus rapide à entraîner, et il s'est avéré que la majorité du temps de calcul pour ce classifieur se passait dans la partie prédiction (environ une minute pour l'entraînement, le reste pour les prédictions, le tout sur 20 minutes en moyenne). Cela le rend donc au final peu performant pour une utilisation réelle, surtout quand on le compare aux réseaux neuronaux de la partie 2 qui peuvent mettre 1h, voire 2h à être entraînés, mais qui ne mettent que quelques secondes à effectuer un grand nombre de prédictions.

- Pour conclure, l'obtention de bons résultats tourne principalement autour de certains aspects :
 - Les réglages effectués sur les classifieurs (taux de convergence, degré du polynome utilisé pour la classification, kernel utilisé, etc.). C'est ce qui a fait la différence sur le SVC
 - Le fonctionnement interne du classifieur, qui peut être adapté (ou pas) aux données traitées. La classification d'images est mieux réalisée ici par le SVC et, dans la partie 2, par les CNN qui possèdent le fonctionnement interne idéal (Convolutions pour les CNNs, gestion efficace des dimensions pour le SVC)
 - La qualité des données : une grande majorité de nos données sont floutées et il est difficile dans ce cas de délimiter le chiffre visé afin de correctement classer. Cependant, des pré-traitements tels que la binarisation et l'égalisation d'histogramme / expansion dynamique permettent de mieux délimiter les données importantes, facilitant ainsi le travail des classifieurs