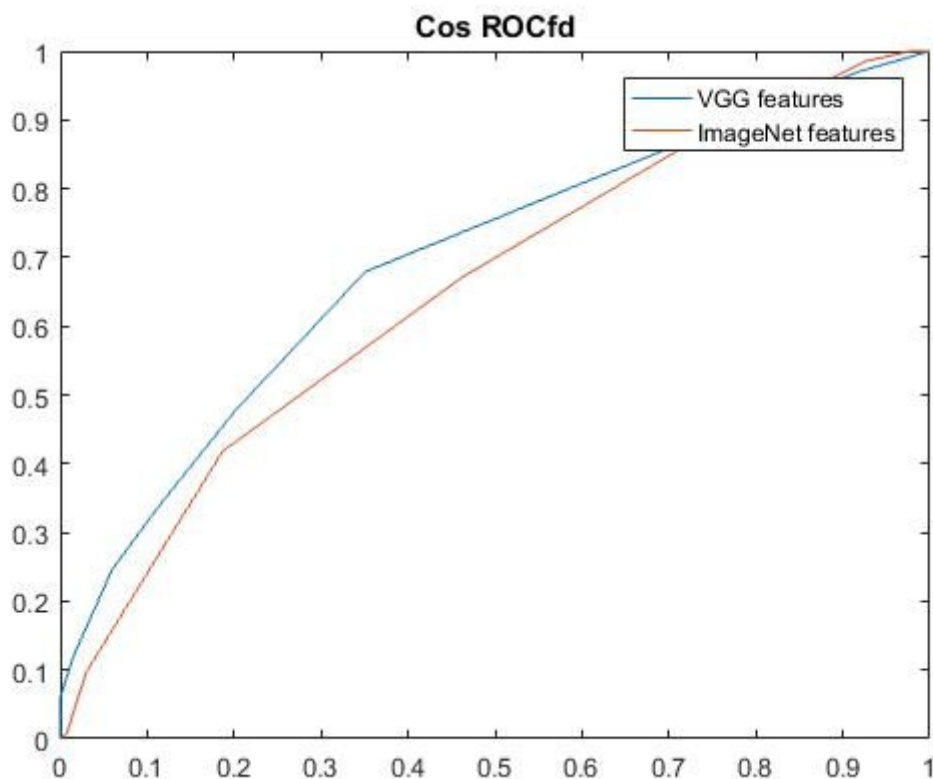


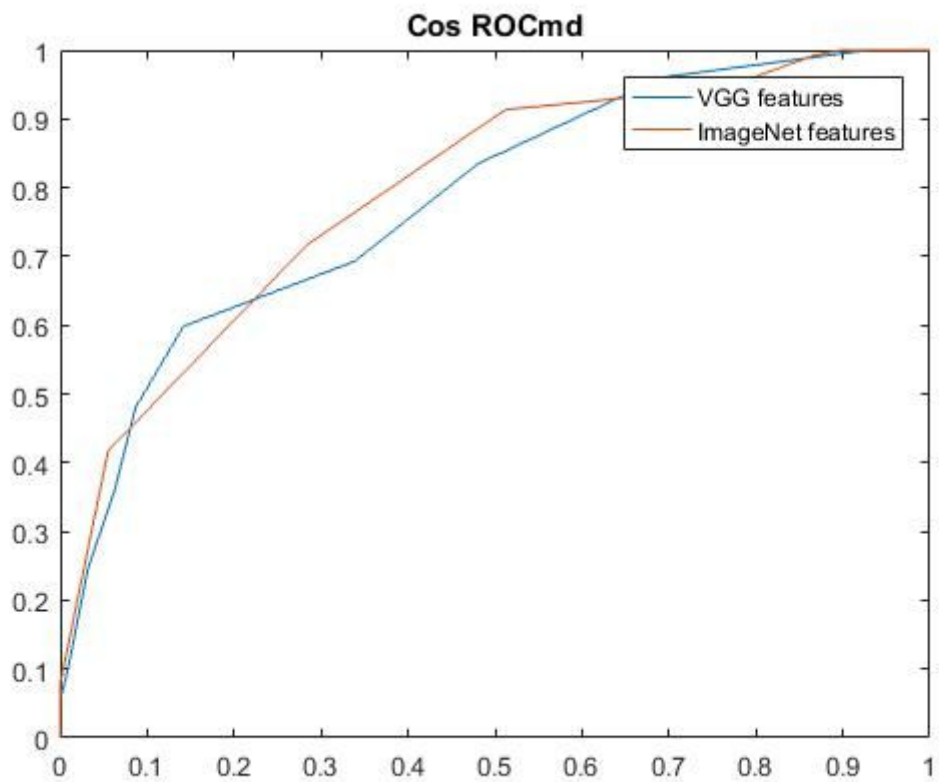
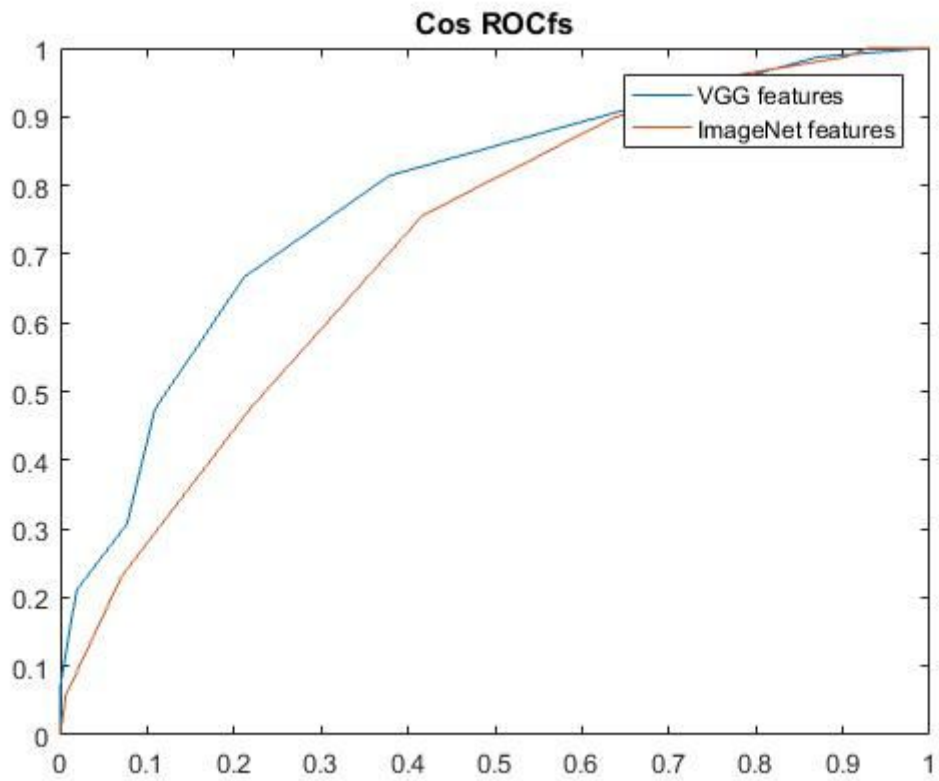
Appendix - Step by step development and results

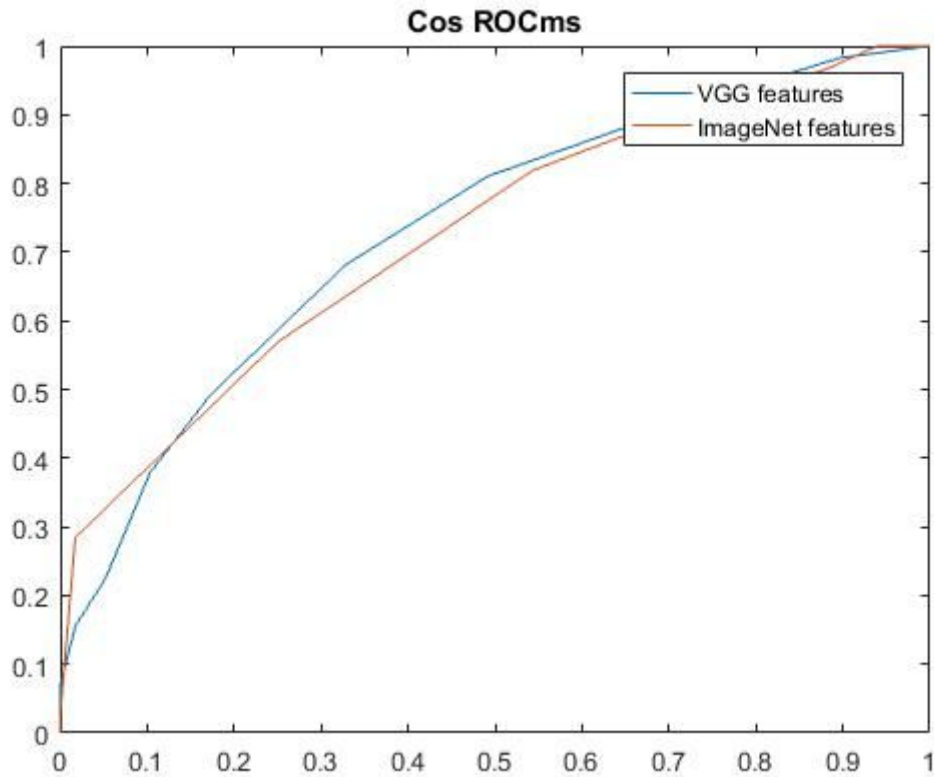
This section will contain all the experiments performed to achieve the final results. The methodology followed in the development has been the following: every component is set as it is and only one component is added/modified at a time. Then, the results for the new setting are computed and compared with the best previous results. Experimentation continues from the setting that obtained the best results. The majority of the experiments were performed in the database KinFace-I because most of the proposals perform worse on this than in KinFace-II. In order to compare results is better sticking to a dataset.

As first step, VGG-Face and VGG-F features were calculated. In order to measure the accuracy of them, ROC curves were calculated for the 4 relationship pairs. In this step, in order to measure the accuracy, cosine similarity was used.

ROC of each pair using cosine similarity. KinFace I







After that, we wanted to measure the accuracy of a real classifier on the features. State of the Art solutions mostly used SVMs because they are fast and work well on big dimension features. Concretely, Gaussian kernel SVMs are the most used, so we tried linear and Gaussian SVMs. See tables 1 and 2

Table 1: Accuracy of deep features with linear SVM classifier. KinFacel

Feature	fd	fs	md	ms	mean
VGG-Face	0.5933	0.6603	0.6535	0.5905	0.6244
VGG-F	0.6082	0.6699	0.7323	0.6638	0.6685

Table 2: Accuracy of deep features using gaussian SVM classifier. KinFacel

Feature	fd	fs	md	ms	mean
VGG-Face	0.5709	0.6122	0.6299	0.5905	0.6008

VGG-F	0.6119	0.6955	0.7598	0.6767	0.6860
-------	--------	--------	--------	--------	--------

We chose to use the SVM classifier with Gaussian kernel because it performed the best results on average (feature VGG-F) and because it is used in most of the state of the art approaches. It is curious to see that even if VGG-Face was trained on faces, VGG-F that was trained in object recognition works better for kinship verification.

In order to create a super classifier combining the Gaussian SVMs of the two deep features, we used the score of the classifiers (The sign of each instance, positive or negative, indicates that it belongs to a class or the other and its absolute value means how far it is from the hyperplane). Each SVM's score is multiplied by a factor (the sum of all the factor equals to 1) and the sum of this coefficients for all the SVMs indicates whether each test instance belongs to one class or the other just looking at the sign. The following table 3 shows VGG-Face and VGG-F Gaussian SVMs blended weighing 1/2 each classifier.

Table 3: Accuracy of deep features by weighting each Gaussian SVM in 1/2. KinFacel

Feature	fd	fs	md	ms	mean
1/numFeatures weighting for blending	0.6269	0.6763	0.7323	0.6681	0.6759

Until now, we have seen that using the the deep features + Gaussian SVM blending works well. To this structure, we added NRML and MNRML projections to see if any of them improves the results. In order to merge NRML SVMs, each classifier is weighted as 1/2. For MNRML SVMs blending, we used the beta values output from MNRML projection function, which indicates the relevance of each feature. In the author's code, they use PCA before projections reduce the dimensionality of the features to reduce the computational time . So we added PCA + MNRML/NRML to the pipeline, using as initial point the best configuration that the authors found: Iterations = 1; number neighbours = 5; PCA dimension = 40.

The pipeline for the tables is the following: deep features extraction -> PCA -> MNRML/NRML projection -> Gaussian SVMs blending. See tables 4 and 5

Table 4: Accuracy of PCA + MNRML + gaussian SVM blending. KinFacel. Parameters setting:

- PCA dimension = 40
- MNRML Iterations = 1
- MNRML number neighbours = 5

Feature	fd	fs	md	ms	mean
beta weighting for blending	0.6679	0.7788	0.7992	0.7500	0.7492

The next table shows the mean beta values created by MNRML for table 4 (used for SVMs' blending)

Table 4': Mean of beta values (of all folds and pair type) for table 4.

Features	VGG-Face	VGG-F
Beta merged	0.4931	0.5068

Table 5: Accuracy of PCA + NRML + gaussian SVM blending. KinFaceL. Parameters setting:

- NRML iterations = 1
- NRML number neighbours = 5
- PCA dimension = 40

Feature	fd	fs	md	ms	mean
1/numFeatures weighting for blending	0.6679	0.7949	0.7874	0.7241	0.7436

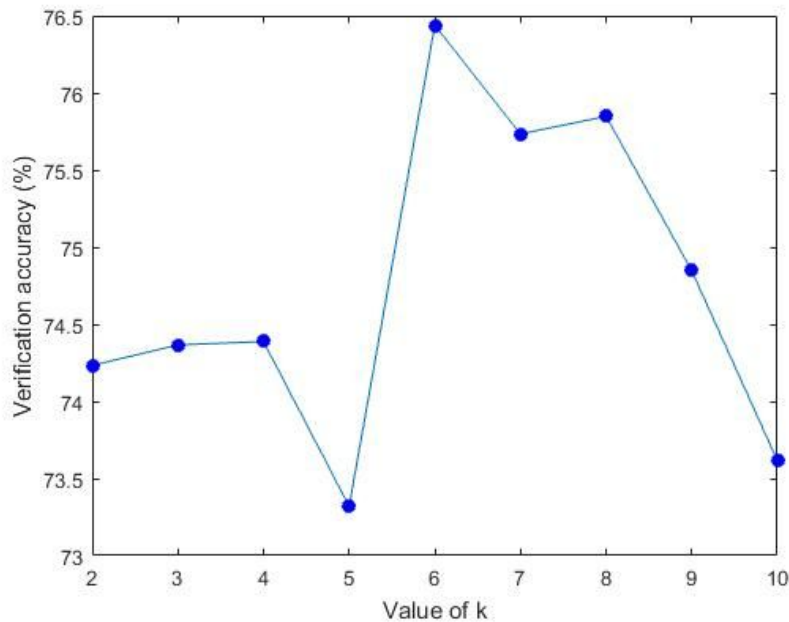
As MNRML gets slightly better results on average and its authors state that MNRML is designed to improve the performance of the features working together, this is the projection method that we chose. Now we wanted to configure its parameters. The methodology used to do so is the following: We set the initial parameters and selected one variable to with which we tried different values in order to find its best value. Then we changed this value in the previous configuration and tried modifying another parameter. This procedure was done until all the parameters were set. We assume that this way, we can find the best (or at least a good combination) of the values of the parameters.

First, we fix the parameters as in the original MNRML paper. These are the values:

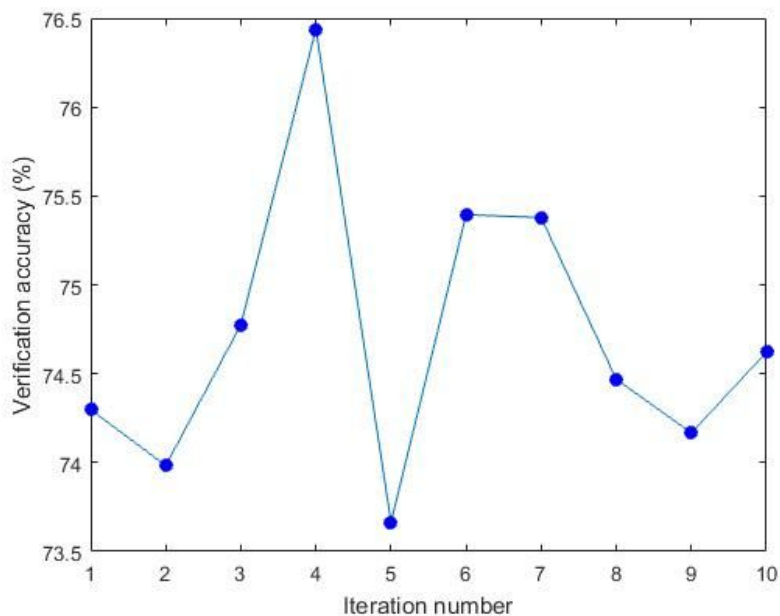
- PCA dimension = 40
- MNRML iterations = 1
- MNRML number neighbours (k) = 5

The accuracy is calculated using the mean for the 4 pairs (father-daughter, father-son, mother-daughter and mother-son)

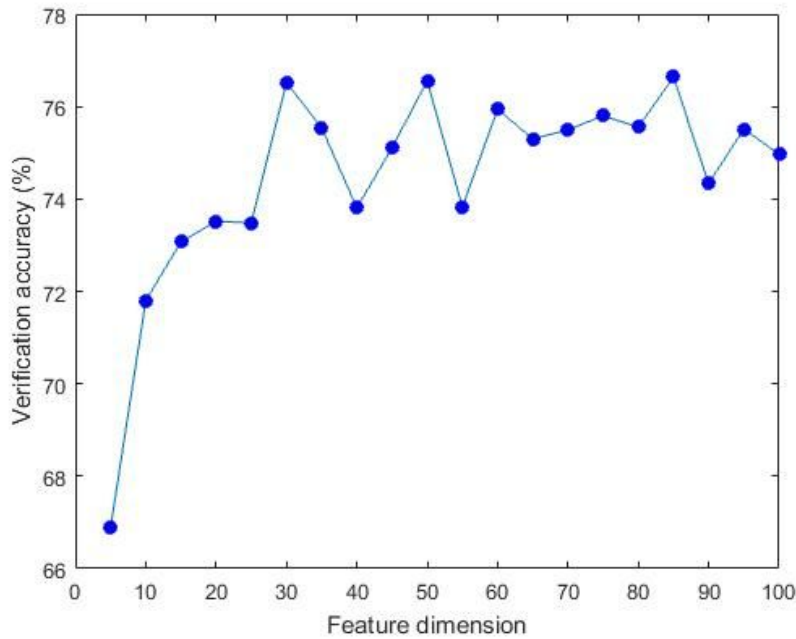
First we try with different values of k (number neighbours). In the figure we can see that the best value of knn is equal to 6.



Now, we set k to 6 and PCA dimension to 30 and change the number of iterations. The figure shows that the best number of iterations is 4.



We finally set the number PCA dimension (making steps so as to reduce computational time):



We can see that changing the feature dimension to more than 30 does not improve the performance, so we set the value to 30.

The best parameter setting found in the last search:

- PCA dimension = 30
- MNRML iterations = 4
- MNRML number neighbours = 6

At this point we were not completely sure that SVMs' best kernel was gaussian rather than linear (this could have changed when added the MNRML projection), so we again tested the current pipeline with different kernels. See tables 6 and 7

Table 6: Accuracy of PCA + MNRML + linear SVM blending. KinFacel.

Feature	fd	fs	md	ms	mean
beta weighting for blending	0.6455	0.7532	0.8031	0.8017	0.7509

Table 7: Accuracy of PCA + MNRML + gaussian SVM blending. KinFacel.

Feature	fd	fs	md	ms	mean
beta weighting for blending	0.6903	0.7821	0.7953	0.7931	0.7652

In order to compare the results with original MNRML paper's, we created table 7', where we have just changed the order of the pairs and converted the number into percentage order to be easier to visually compare the results

Table 7': Accuracy of that moment developed pipeline in KinFacel.

Feature	F-S	F-D	M-S	M-D	mean
beta weighting for blending	78.2	69.0	79.3	79.5	76.5

The following figure shows MNRML paper's results

TABLE 4
Classification Accuracy (Percent) of Different Methods
on Different Subsets of the KinFaceW-I Data Set

Method	Feature	F-S	F-D	M-S	M-D	Mean
CSML	LBP	63.7	61.2	55.4	62.4	60.7
	LE	61.1	58.1	60.9	70.0	62.5
	SIFT	66.5	60.0	60.0	56.4	59.8
	TPLBP	57.3	61.5	63.2	57.0	59.7
NCA	LBP	61.7	62.2	56.4	62.4	60.7
	LE	62.1	57.1	61.9	69.0	62.3
	SIFT	67.5	61.0	61.0	57.4	60.8
	TPLBP	56.3	60.5	62.2	56.0	58.7
LMNN	LBP	62.7	63.2	57.4	63.4	61.7
	LE	63.1	58.1	62.9	70.0	63.3
	SIFT	69.5	63.0	63.0	59.4	62.8
	TPLBP	57.3	61.5	63.2	57.0	59.7
NRML	LBP	64.7	65.2	59.4	65.4	63.7
	LE	64.1	59.1	63.9	71.0	64.3
	SIFT	70.5	64.0	64.0	60.4	63.8
	TPLBP	59.3	63.5	65.2	60.0	62.9
MNRML	All	72.5	66.5	66.2	72.0	69.9

Now that we obtained better results than MNRML paper in KinFace-I database, we wanted perform several tests on the KinFace-II database to test that the pipeline works well also in this database. The results are in tables 8, 9, 10 and 11.

Table 8: Accuracy of deep features with gaussian SVM classifier. KinFacell. Comparable to table table 2

Feature	fd	fs	md	ms	mean
VGG-Face	0.5860	0.5660	0.6140	0.5760	0.5855
VGG-F	0.7460	0.7620	0.8060	0.7800	0.7735

Table 9: Accuracy of deep features by weighing each Gaussian SVM in $\frac{1}{2}$. KinFacell. Comparable to table 3

Feature	fd	fs	md	ms	mean
1/numFeatures weighting for blending	0.7420	0.7720	0.7900	0.7540	0.7645

Table 10: Accuracy of PCA + MNRML + gaussian SVM blending. KinFacell. Comparable to table 6. Parameters setting:

- PCA dimension = 30
- MNRML iterations = 4
- MNRML number neighbours = 6

Feature	fd	fs	md	ms	mean
beta weighting for blending	0.7640	0.8000	0.7880	0.8280	0.7950

Table 11: Accuracy of PCA + NRML + gaussian SVM blending. KinFacell. Parameters setting:

- PCA dimension = 30
- NRML iterations = 4
- NRML number neighbours = 6

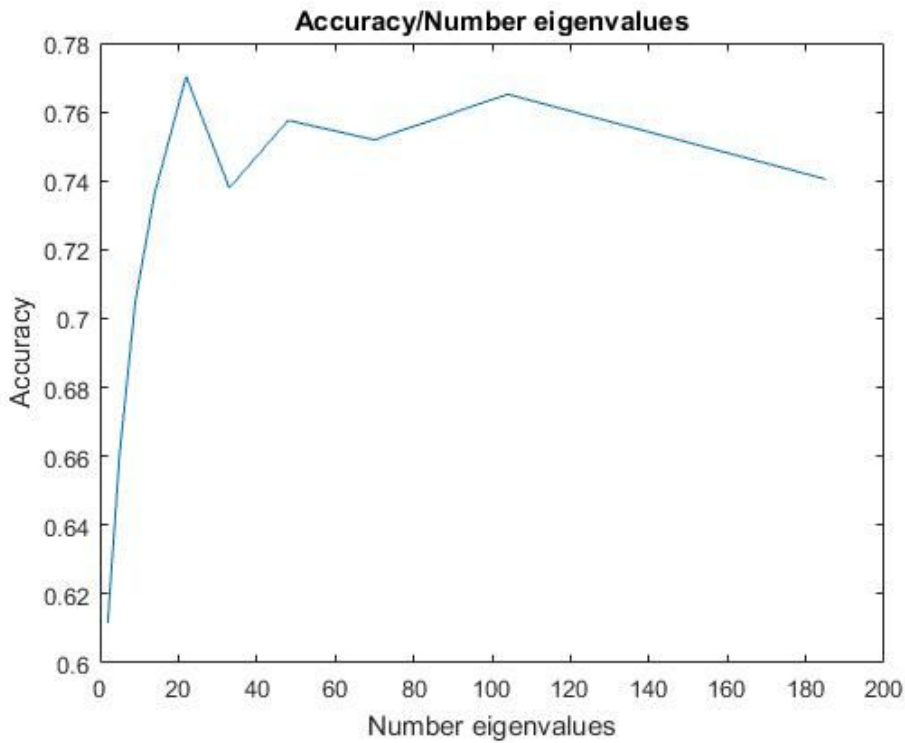
Feature	fd	fs	md	ms	mean
1/numFeatures weighting for blending	0.7360	0.8120	0.8220	0.8060	0.7940

The results of NRML are quite similar to the ones obtained by MNRML. So as in KinFace-I, we chose to stick to MNRML.

At this point, we have seen that databases KinFacel and KinFacell score similarly with regard to different classification pipelines, so we decided to switch back to the KinFace-I to perform the experiments because this is more difficult than KinFace-II. Sticking to a database also makes easier the results comparison.

We wanted to improve the results of KinFace-I, so we decided to make an exhaustive search to find the best PCA dimension value. See the next figure

Plot of PCA+MNRML+SVM blending mean accuracy when changing PCA dimension (number of selected eigenvectors). KinFace-I



The plot shows that the best number of eigenvalues is 27. From now on, PCA dimension is set to 27.

Now, the pipeline’s accuracy is recalculated using the new pca dimension value. See table 12

Table 12: Accuracy of PCA + MNRML + gaussian SVM blending. KinFaceI. Parameters setting:

- **PCA dimension = 27**
- MNRML Iterations = 4
- MNRML number neighbours = 6

Feature	fd	fs	md	ms	Mean
beta weighting for blending	0.7089	0.8108	0.7716	0.7629	0.7636

Another component that can be added to the pipeline is LDE projection. The next table shows previous pipeline but adding LDE. LDE’s K1 and K2 parameters have been set by performing exhaustive search for all K1 in range 1:10 and K2 in range 1:10

Table 13: Accuracy of PCA + MNRML + LDE + gaussian SVM blending. KinFaceI. Parameters setting:

- PCA dimension = 27
- MNRML Iterations = 4
- MNRML number neighbours = 6
- **LDE K1 = 6**
- **LDE K2 = 2**

Feature	fd	fs	md	ms	Mean
beta weighting for blending	0.7238	0.8108	0.7952	0.7456	0.7689

The results show that LDE improve the results in KinFace-I in 0.5 % on average. It is not a significant improvement because there is still a lot to improve.

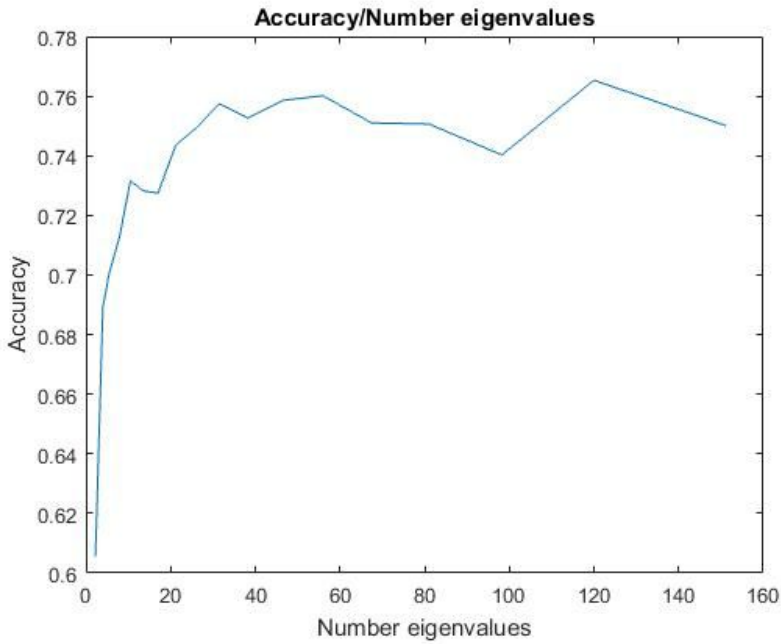
In this point we decided to add more features with the hope that they give complementary information to the current deep features. Concretely, the new features added are traditional LBP and HOG.

The next plot shows how the pipeline (LBP+HOG+VGG-Face+VGG-F) + PCA + MNRML + gaussian SVMs perform when changing the PCA dimension, as it has shown to be an important part of the pipeline. LBP and HOG were configured the following way (default configuration used in master's Computer Vision subject):

- LBP: number of neighbors 16 and radius 2
- HOG: cell size 19x19 and number of bins 12

KinFace-I with deep learning + traditional features

Plot of the pipeline (LBP+HOG+VGG-Face+VGG-F) + PCA + MNRML + gaussian SVMs when changing the PCA dimension (number of eigenvalues)



We also wanted to measure the performance of each feature individually when using gaussian SVM as classifier. This is shown in table 14

Table 14: Accuracy of VGG-Face+VGG-F+LBP+HOG features using gaussian SVM classifier. KinFacel. Comparable to table 2

- LBP: number of neighbors 16 and radius 2
- HOG: cell size 19x19 and number of bins 12

Feature	fd	fs	md	ms	mean
VGG-Face	0.5709	0.6122	0.6299	0.5905	0.6244
VGG-F	0.6119	0.6955	0.7598	0.6767	0.6685
LBP	0.5933	0.6122	0.5748	0.5690	0.5873
HOG	0.6082	0.6506	0.6417	0.6207	0.6303

We tried to improve the results of LBP and HOG individually by tweaking their parameters. We looked at the best parameter configuration found for similar size images. The chosen configuration is the following:

- LBP: number of neighbors 16 and radius 4
- HOG: cell size 15x15 and number of bins 13

Now we execute the same experiment of table 14 but with the new configuration. See table 15

Table 15: Accuracy of VGG-Face+VGG-F+LBP+HOG features using gaussian SVM classifier. KinFaceI. Same as previous Table but traditional features configured as Huerta et al. paper:

- LBP: number of neighbors 16 and radius 4
- HOG: cell size 15x15 and number of bins 13

Feature	fd	fs	md	ms	mean
VGG-Face	0.5709	0.6122	0.6299	0.5905	0.6244
VGG-F	0.6119	0.6955	0.7598	0.6767	0.6685
LBP	0.6082	0.5833	0.5827	0.5905	0.5912
HOG	0.6231	0.7051	0.6339	0.6078	0.6424

Once again, the results of MNRML paper are shown to compare the performance of each feature individually. It seems that our features are better on average.

TABLE 4
Classification Accuracy (Percent) of Different Methods on Different Subsets of the KinFaceW-I Data Set

Method	Feature	F-S	F-D	M-S	M-D	Mean
CSML	LBP	63.7	61.2	55.4	62.4	60.7
	LE	61.1	58.1	60.9	70.0	62.5
	SIFT	66.5	60.0	60.0	56.4	59.8
	TPLBP	57.3	61.5	63.2	57.0	59.7
NCA	LBP	61.7	62.2	56.4	62.4	60.7
	LE	62.1	57.1	61.9	69.0	62.3
	SIFT	67.5	61.0	61.0	57.4	60.8
	TPLBP	56.3	60.5	62.2	56.0	58.7
LMNN	LBP	62.7	63.2	57.4	63.4	61.7
	LE	63.1	58.1	62.9	70.0	63.3
	SIFT	69.5	63.0	63.0	59.4	62.8
	TPLBP	57.3	61.5	63.2	57.0	59.7
NRML	LBP	64.7	65.2	59.4	65.4	63.7
	LE	64.1	59.1	63.9	71.0	64.3
	SIFT	70.5	64.0	64.0	60.4	63.8
	TPLBP	59.3	63.5	65.2	60.0	62.9
MNRML	All	72.5	66.5	66.2	72.0	69.9

Table 15 shows the performance of the 4 features with the pipeline

Table 16: Accuracy of (VGG-Face+VGG-F+LBP+HOG)+PCA + MNRML + gaussian SVM blending. KinFacel. Comparable to table 12

- PCA dimension = 27
- MNRML Iterations = 4
- MNRML number neighbours = 6

Feature	fd	fs	md	ms	mean
beta weighting for blending	0.6865	0.7756	0.7677	0.7974	0.7568

The next table shows the mean beta values created by MNRML for table 16 (used for SVMs' blending)

Table 16': Mean of beta values (of all folds and pair type) for table 16.

Features	VGG-Face	VGG-F	LBP	HOG
Beta merged	0.2361	0.2428	0.2695	0.2517

Now we tested adding LDE to the previous table's setup, in table 16. LDE's K1 and K2 parameters have been set by performing exhaustive search for all K1 in range 1:10 and K2 in range 1:10

Table 17: Accuracy of (VGG-Face+VGG-F+LBP+HOG) + PCA + MNRML + LDE + gaussian SVM blending. KinFacel. Comparable to table 13. Parameters setting:

- PCA dimension = 27
- MNRML Iterations = 4
- MNRML number neighbours = 6
- **LDE K1 = 5**
- **LDE K2 = 8**

Feature	fd	fs	md	ms	Mean
beta weighting for blending	0.7015	0.7628	0.7795	0.7586	0.7506

In order to see whether adding LBP and HOG traditional features improve the results, the next table 18 computes the same pipeline as table 17, but this time only for VGG-Face and VGG-F deep features

Table 18: Accuracy of (VGG-Face+VGG-F) + PCA + MNRML + LDE + gaussian SVM blending. KinFace1. Comparable to table 17. Parameters setting:

- PCA dimension = 27
- MNRML Iterations = 4
- MNRML number neighbours = 6
- LDE K1 = 5
- LDE K2 = 8

Feature	fd	fs	md	ms	Mean
beta weighting for blending	0.7089	0.8013	0.7756	0.7414	0.7568

Apparently, adding traditional features does not improve the results (the traditional features perform worse than deep features) but does not make it much worse either. The conclusion gained from this table is that LBP and HOG do not provide extra information to the one that VGG-Face and VGG-F provide together, so from this point on, only the deep features are used.

The next experiment aims to test the pipeline (VGG-Face+VGG-F) + MNRML + LDE + gaussian SVM blending, to test the performance of the pipeline without PCA. The results are in the table 19. Only father-daughter relationship has been calculated because it takes much time to process. However, we can see that the accuracy is around 10% worse, so PCA improves the results; apart from reducing considerably the processing time.

Table 19: Accuracy of MNRML projected features (VGG-Face+VGG-F) + LDE (without PCA). KinFacel.

Parameters setting:

- PCA dimension = 27
- MNRML Iterations = 4
- MNRML number neighbours = 6
- LDE K1 = 5
- LDE K2 = 8

Feature	fd
beta weighting for blending	0.6007

Table 20 shows the same processing using traditional and deep features.

Table 20: Accuracy of MNRML projected features (VGG-Face+VGG-F+LBP+HOG) + LDE (without PCA).

KinFacel. Parameters setting:

- PCA dimension = 27
- MNRML Iterations = 4
- MNRML number neighbours = 6
- LDE K1 = 5
- LDE K2 = 8

Feature	fd	fs	md	ms	Mean
beta weighting for blending	0.6119	0.5833	0.5787	0.5733	0.5868

This table confirms the same as table 19: the accuracy without PCA is around 10% worse, so PCA improves the results; apart from reducing considerably the processing time.

The next experiment consists on optimizing the PCA dimension per each pair type. After an exhaustive search, the results are recorded in table 21.

Table 21: Accuracy of (VGG-Face+VGG-F+LBP+HOG) + PCA + MNRML + LDE + gaussian SVM blending. KinFacel. PCA dimension optimized per each pair. Parameters setting:

- PCA dimension = [26 43 39 37]
- MNRML Iterations = 4
- MNRML number neighbours = 6
- LDE K1 = 5
- LDE K2 = 8

Feature	fd	fs	md	ms	Mean
beta weighting for blending	0.7351	0.7981	0.8071	0.7845	0.7812

The mean accuracy has improved.

The next table shows the mean beta values created by MNRML for table 21 (used for SVMs' blending)

Table 21': Mean of beta values (of all folds and pair type) for table 21.

Features	VGG-Face	VGG-F	LBP	HOG
beta weighting for blending	0.2348	0.2435	0.2699	0.2518

The following table shows the experiment of previous table (21) but changing the way of merging the SVMs

Table 22: Accuracy of (VGG-Face+VGG-F+LBP+HOG) + PCA + MNRML + LDE + gaussian SVM blending. KinFacel. Way of merging: assigning relevance = $1/\text{numFeat}$ to each feature's SVM. Parameters setting:

- PCA dimension = [26 43 39 37]

- MNRML Iterations = 4
- MNRML number neighbours = 6
- LDE K1 = 5
- LDE K2 = 8

Feature	fd	fs	md	ms	Mean
1/numFeatures weighting for blending	0.7425	0.8109	0.8071	0.7629	0.7808

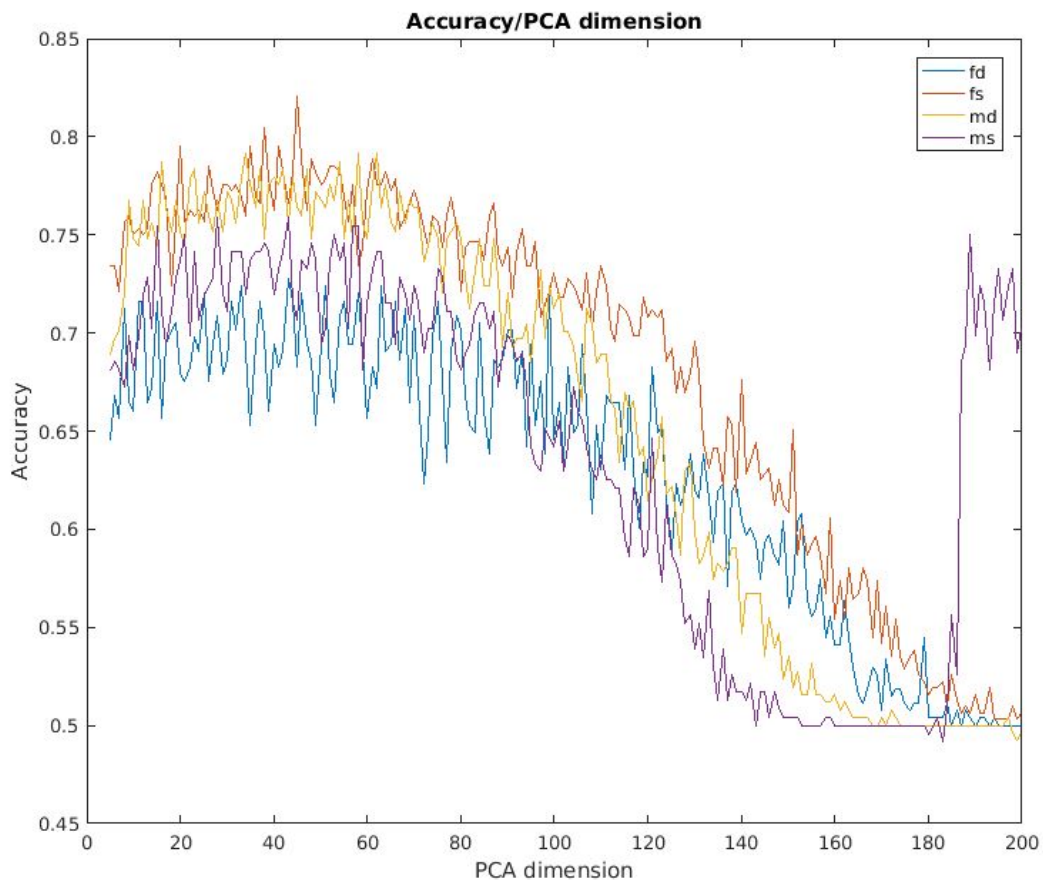
Last experiment shows that using betas for scoring each features relevance when blending SVMs or using 1/number_of_features is very similar. We chose to maintain the betas.

The next table shows the weighting used for each feature for table 22 (used for SVMs' blending)

Table 22': Mean of beta values (of all folds and pair type) for table 22

Features	VGG-Face	VGG-F	LBP	HOG
beta weighting for blending	0.25	0.25	0.25	0.25

The next figure shows how the performance of each pair changes when changing the PCA dimension for the four pairs in the following pipeline: (LBP+HOG+VGG-Face+VGG-F) + PCA + MNRML + LDE + gaussian SVMs



There are two things to see in the figure. The first one is that the accuracy of the classifier decreases with the dimension (from 40 onwards). The second one is that for mother-son pair, the accuracy decreases and from 180 on it increases.

We decided to test the until now successful pipeline to see the impact of each component in the accuracy. The results are in tables 23 to 28.

Table 23: Accuracy of (VGG-Face+VGG-F+LBP+HOG)+PCA + MNRML + gaussian SVM blending (no LDE). KinFacel.

Parameter setting:

- PCA dimension = [26 43 39 37]
- MNRML iterations = 4
- MNRML number neighbours = 6

Feature	fd	fs	md	ms	Mean
beta weighting for blending	0.6978	0.7949	0.7874	0.7629	0.7607

This results can seem worse but they do not have the PCA dimension optimized for this setting, so they are good results.

The table 24 shows the results of removing the MNRML projection. they are worse than with MNRML.

Table 24: (VGG-Face+VGG-F+LBP+HOG) + PCA + LDE + gaussian SVM blending (no MNRML). KinFacel. Parameter setting:

- PCA dimension = [26 43 39 37]
- LDE K1 = 5
- LDE K2 = 8
- Feature weighting = 1/number_of_features

Feature	fd	fs	md	ms	Mean
1/numFeatures weighting for blending	0.7127	0.7788	0.7795	0.7284	0.7499

Table 25 shows the results of fusing the features with SVM, removing the rest of the pipeline

Table 25: (VGG-Face+VGG-F+LBP+HOG) + gaussian SVM blending (no PCA, no MNRML, no LDE). KinFacel. Parameter setting:

- Feature weighting = 1/number_of_features

Feature	fd	fs	md	ms	Mean
1/numFeatures weighting for blending	0.6455	0.7115	0.7480	0.6638	0.6922

Table 26 shows the results of applying PCA and removing MNRML and LDE from the pipeline.

Table 26: (VGG-Face+VGG-F+LBP+HOG) + PCA + gaussian SVM blending (no MNRML, no LDE). KinFace1. Parameter setting:

- PCA dimension = [26 43 39 37]
- Feature weighting = 1/number_of_features

fd	fs	md	ms	Mean
0.6866	0.7756	0.7756	0.7328	0.7427

Table 27 shows the results of only applying MNRML and removing PCA and LDE from the pipeline.

Table 27: (VGG-Face+VGG-F+LBP+HOG) + MNRML+ gaussian SVM blending (no PCA, no LDE). KinFace1. Parameter setting:

- MNRML iterations = 4
- MNRML number neighbours = 6
- Feature weighting = 1/number_of_features

fd	fs	md	ms	Mean
0.6007	0.5994	0.5945	0.6164	0.6027

Table 28 shows the results of only applying LDE and removing PCA and MNRML from the pipeline.

Table 28: Accuracy of LDE fused SVMs (deep+traditional) (no PCA, no MNRML)

- LDE K1 = 5
- LDE K2 = 8
- Feature weighting = 1/number_of_features

fd	fs	md	ms	Mean
0.6194	0.7468	0.6654	0.6509	0.6706

We can see that each part is important for the classification. However, LDE does not improve significantly (the pipeline scores 2% worse without LDE but that is because we have not optimized the PCA dimension per type of kinship relationship for the pipeline without LDE. After the experiments 23-28, we decided to remove the LDE projection from the pipeline. However, this decision will not be finally implemented until later experiments.

In this point we saw that PCA was an important component of the pipeline, so we decided to try different feature selections methods; also because the deep features have a high dimension so there can be noisy descriptors that do not give information about the class and confuse the classifier.

In the tables 29-34, the variance threshold has been used for feature selection, which removes the descriptors that have a variance less than or equal to the threshold. In table 29, all the features have been filtered by the threshold but in tables 30-34 only deep features have been filtered because the when the variance filter with these values make LBP and HOG traditional features' descriptors have a size of 0 (disappear).

Table 29: Accuracy of (VGG-Face+VGG-F+LBP+HOG) + Variance feature selection + PCA + MNRML + LDE + gaussian SVM blending. KinFace1. PCA dimension optimized for this setting per category (by exhaustive search in range 17-38). Parameters setting:

- Variance threshold = 0
- LDE K1 = 5
- LDE K2 = 8
- Feature weighting = 1/number_of_features

Feature	fd	fs	md	ms	Mean
1/numFeatures weighting for blending	0.7127	0.8045	0.8150	0.7586	0.7727
Size VGG-Face	4086	4083	4043	4059	
Size VGG-F	3494	3604	3474	3470	
Size LBP	243	243	243	243	
Size HOG	468	468	468	468	

Table 30: Accuracy of (VGG-Face+VGG-F+LBP+HOG) + Variance feature selection (only to VGG-Face and CGG-F) + PCA + MNRML + LDE + gaussian SVM blending. KinFacel. PCA dimension optimized for this setting per category (by exhaustive search in range 17-38). Parameters setting:

- Variance threshold = 0.1
- LDE K1 = 5
- LDE K2 = 8
- Feature weighting = 1/number_of_features

Feature	fd	fs	md	ms	Mean
1/numFeatures weighting for blending	0.7164	0.8077	0.7992	0.7629	0.7716
Size VGG-Face	3940	3876	3819	3772	
Size VGG-F	1896	1981	1902	1913	
Size LBP	243	243	243	243	
Size HOG	468	468	468	468	

Table 31: Accuracy of (VGG-Face+VGG-F+LBP+HOG) + Variance feature selection (only to VGG-Face and CGG-F) + PCA + MNRML + LDE + gaussian SVM blending. KinFacel. PCA dimension optimized for this setting per category (by exhaustive search in range 17-38). Parameters setting:

- Variance threshold = 0.2
- LDE K1 = 5
- LDE K2 = 8
- Feature weighting = 1/number_of_features

Feature	fd	fs	md	ms	Mean
1/numFeatures weighting	0.7276	0.7917	0.7953	0.7629	0.7694

for blending					
Size VGG-Face	3804	3742	3696	3544	
Size VGG-F	1569	1627	1566	1592	
Size LBP	243	243	243	243	
Size HOG	468	468	468	468	

Table 32: Accuracy of (VGG-Face+VGG-F+LBP+HOG) + Variance feature selection (only to VGG-Face and CGG-F) + PCA + MNRML + LDE + gaussian SVM blending. KinFaceL. PCA dimension optimized for this setting per category (by exhaustive search in range 17-38). Parameters setting:

- Variance threshold = 0.5
- LDE K1 = 5
- LDE K2 = 8
- Feature weighting = 1/number_of_features

Feature	fd	fs	md	ms	Mean
1/numFeatures weighting for blending	0.7239	0.7949	0.8071	0.7716	0.7743
Size VGG-Face	3434	3286	3348	3033	
Size VGG-F	1051	1122	1030	1077	
Size LBP	243	243	243	243	
Size HOG	468	468	468	468	

Table 33: Accuracy of (VGG-Face+VGG-F+LBP+HOG) + Variance feature selection (only to VGG-Face and CGG-F) + PCA + MNRML + LDE + gaussian SVM blending. KinFacel. PCA dimension optimized for this setting per category (by exhaustive search in range 17-38). Parameters setting:

- Variance threshold = 1
- LDE K1 = 5
- LDE K2 = 8
- Feature weighting = 1/number_of_features

Feature	fd	fs	md	ms	Mean
1/numFeatures weighting for blending	0.7201	0.7949	0.7953	0.7672	0.7694
Size VGG-Face	2883	2637	2856	2408	
Size VGG-F	543	566	527	553	
Size LBP	243	243	243	243	
Size HOG	468	468	468	468	

Table 34: Accuracy of (VGG-Face+VGG-F+LBP+HOG) + Variance feature selection (only to VGG-Face and CGG-F) + PCA + MNRML + LDE + gaussian SVM blending. KinFacel. PCA dimension optimized for this setting per category (by exhaustive search in range 17-38). Parameters setting:

- Variance threshold = 2
- LDE K1 = 5
- LDE K2 = 8
- Feature weighting = 1/number_of_features

Feature	fd	fs	md	ms	Mean
1/numFeatures weighting	0.7388	0.7949	0.7953	0.7586	0.7719

for blending					
Size VGG-Face	2000	1722	2120	1577	
Size VGG-F	103	110	96	90	
Size LBP	243	243	243	243	
Size HOG	468	468	468	468	

Tables 29-34 show that variance feature selection does not improve the results but reducing the dimensionality of deep features is possible without affecting the performance of the pipeline.

The next two tables make experiments to compare the performance of the full pipeline for the 4 features and only deep features, in order to see which one gets better results. The feature selection method has been replaced by the fisher feature selection. For calculating the score, subtraction of pairs' features is performed and then the absolute value of this is calculated.

Table 35: Accuracy of (VGG-Face+VGG-F+LBP+HOG) + Fisher feature selection with subtraction + PCA + MNRML + LDE + gaussian SVM blending. KinFacel. PCA dimension optimized for this setting per category (by exhaustive search in range 15-70). Parameters setting:

- Fisher dimension reduction = 0.5
- PCA dimension = [47 59 55 40]
- LDE K1 = 5
- LDE K2 = 8

Feature	fd	fs	md	ms	Mean
beta weighting for blending	0.7724	0.8333	0.8031	0.7845	0.7983

Table 36: Accuracy of (VGG-Face+VGG-F) + Fisher feature selection with subtraction + PCA + MNRML + LDE + gaussian SVM blending. KinFacel. PCA dimension optimized for this setting per category (by exhaustive search in range 15-70). Parameters setting:

- Fisher dimension reduction = 0.5
- PCA dimension = [39 34 38 18]
- LDE K1 = 5
- LDE K2 = 8

Feature	fd	fs	md	ms	Mean
beta weighting for blending	0.7425	0.8141	0.8346	0.7888	0.7950

The conclusion for tables 35 and 36 is that using LBP and HOG features do not improve the results, so in this point, we decided to remove LBP and HOG features and stick only to VGG-Face and VGG-F deep features.

The next table 37 aims to finally see whether LDE improves the pipeline's performance.

Table 37: Accuracy of (VGG-Face+VGG-F) + Fisher feature selection with subtraction + PCA + MNRML + gaussian SVM blending (no LDE). KinFacel. PCA dimension optimized for this setting per category (by exhaustive search in range 15-70). Parameters setting:

- Fisher dimension reduction = 0.5
- PCA dimension = [15 52 60 55]
- LDE K1 = 5
- LDE K2 = 8

Feature	fd	fs	md	ms	Mean
beta weighting for blending	0.7575	0.8301	0.8425	0.8017	0.8080

Comparing table 37 with 36, we can see that adding LDE to the pipeline does not improve its accuracy.

Summary of the conclusions of tables 35-37

1. Traditional features do not help at all, so it is enough with deep features. LDE does not improve either, could be because data after Fisher+PCA+MNRML is very compressed so no improvement is made using LDE.
2. Fisher feature selection with subtraction (supervised) is better than variance feature selection (unsupervised)
3. **Fisher feature selection improves results around 2%**

From this point on, we decided to only keep deep features (VGG-Face and VGG-F), remove LDE from the pipeline and continue testing with supervised feature selection methods.

We decided to test the best configuration until that point by tweaking the feature selection parameters in table 38

Table 38: Accuracy of (VGG-Face+VGG-F) + Fisher feature selection with subtraction + PCA + MNRML + gaussian SVM blending. KinFacel. PCA dimension optimized for this setting per category (by exhaustive search in range 15-70).

Parameters setting:

- SVM's Beta merging

Fisher dimension reduction	PCA dim	fd	fs	md	ms	Mean
0.5	15 52 60 55	0.7575	0.8301	0.8425	0.8017	0.8080
0.4	52 52 63 68	0.7649	0.8333	0.8425	0.8233	0.8160
0.3	20 38 44 51	0.7575	0.8205	0.8386	0.8233	0.8100
0.2	35 66 43 68	0.7687	0.8237	0.8504	0.8362	0.8197
0.1	29 58 35 59	0.7649	0.8301	0.8504	0.8707	0.8290
0.05	37 36 64 54	0.7649	0.8109	0.8425	0.8448	0.8158

Table 38, concretely the row of fisher dimension = 0.1 has the best results obtained in the whole set of experiments until now.

The next experiment is changing the fisher feature selection method's pair merging technique. Table 39 shows the results of the pipeline using the fisher feature selection method with concatenation (concatenating parent's and child's feature and selecting a set best percentage for each of them, which are the descriptors selected for the pair).

Table 39: Accuracy of (VGG-Face+VGG-F) + Fisher feature selection with concatenation + PCA + MNRML + gaussian SVM blending. KinFace-I. PCA dimension optimized for this setting per category (by exhaustive search in range 15-70).

Parameters setting:

- SVM's Beta merging

Fisher dimension reduction	PCA dim	fd	fs	md	ms	Mean
0.3 (max 0.6)	54 55 66 41	0.7239	0.8045	0.8307	0.7888	0.7870
0.2 (max 0.4)	47 69 57 52	0.7239	0.7981	0.8307	0.7716	0.7811
0.1 (max 0.2)	64 65 52 63	0.7090	0.7917	0.8307	0.7543	0.7714
0.05 (max 0.1)	38 35 38 69	0.7276	0.7885	0.8110	0.7759	0.7757
0.025 (max 0.05)	70 65 66 58	0.6791	0.7724	0.7717	0.7543	0.7444
0.01 (max 0.02)	39 31 44 23	0.6791	0.7372	0.7283	0.7026	0.7118

Comparing table 39 against table 38, we can deduce that fisher feature selection with subtraction works better than fisher feature selection with concatenation, at least for KinFace-I database. We expect to happen the same in KinFace-II database. Let's test it in tables 40 and 41.

Table 40: Accuracy of (VGG-Face+VGG-F) + Fisher feature selection with subtraction + PCA + MNRML + gaussian SVM blending. KinFacell. PCA dimension optimized for this setting per category (by exhaustive search in range 15-70).

Parameters setting:

- SVM's Beta merging

Fisher dimension reduction	PCA dim	fd	fs	md	ms	Mean
0.5	48 33 38 57	0.7960	0.8520	0.8460	0.8500	0.8360
0.4	30 58 69 67	0.7980	0.8420	0.8560	0.8540	0.8375
0.3	58 64 56 28	0.8180	0.8520	0.8600	0.8600	0.8475
0.2	70 46 61 38	0.8000	0.8600	0.8620	0.8720	0.8485
0.1	45 29 65 31	0.8100	0.8580	0.8760	0.8680	0.8530
0.075	59 47 60 47	0.8080	0.8700	0.8820	0.8700	0.8575
0.05	64 47 61 51	0.8000	0.8580	0.8900	0.8660	0.8535
0.03	26 26 41 65	0.7860	0.8420	0.8780	0.8660	0.8430
0.025	55 69 52 58	0.7900	0.8440	0.8720	0.8580	0.8410
0.02	35 48 57 56	0.7880	0.8380	0.8760	0.8700	0.8430
0.0125	20 50 27 36	0.7620	0.8180	0.8640	0.8400	0.8210

Table 41: Accuracy of (VGG-Face+VGG-F) + Fisher feature selection with concatenation + PCA + MNRML + gaussian SVM blending. KinFacell. PCA dimension optimized for this setting per category (by exhaustive search in range 15-70).

Parameters setting:

- SVM's Beta merging

Fisher dimension reduction	PCA dim	fd	fs	md	ms	Mean
0.5 (max 1)	48 55 59 49	0.7820	0.8320	0.8380	0.8400	0.8230
0.4 (max 0.8)	54 64 70 66	0.7840	0.8360	0.8480	0.8400	0.8270
0.3 (max 0.6)	38 63 58 58	0.7780	0.8440	0.8440	0.8420	0.8270
0.2 (max 0.4)	62 43 61 55	0.7800	0.8260	0.8300	0.8320	0.8170
0.1 (max 0.2)	54 56 38 50	0.7800	0.8160	0.8220	0.8140	0.8080
0.05 (max 0.1)	31 54 47 60	0.7540	0.8040	0.8080	0.8040	0.7925
0.025 (max 0.05)	32 70 63 57	0.7540	0.7780	0.8040	0.7700	0.7765

Conclusion for tables 38-39 and 40-41: Merge with subtraction works better than merging with concatenation for fisher feature selection.

In this moment, we analyzed the impact of each feature selection method on each feature classified individually (no MNRML neither SMV's blending). Results are in table 42

Table 42: Experiments on VGG-F feature

Experiment	fd	fs	md	ms	Mean
VGG-F raw -> SVM	0.6119	0.6955	0.7598	0.6767	0.6860
VGG-F raw -> fisher selection merge: 0.1 ->	0.6418	0.7724	0.7835	0.7716	0.7423

PCA dims = [32 23 58 16]; -> SVM					
VGG-F raw -> fisher selection merge: 0.1 -> SVM	0.6716	0.7276	0.7874	0.7586	0.7363
VGG-F raw -> PCA dims = [45 61 21 25] -> SVM	0.5858	0.7340	0.7756	0.6853	0.6952

The results of table 42 only confirm previous thoughts, that fisher selection using subtractoin + PCA obtain the best accuracy for a feature classification individually.

The last attempt to improve the accuracy of the pipeline is the following: before performing fisher feature selection and after merging the pairs' data by using subtraction, centering the data (subtracting by the mean and dividing by the standard deviation). See table 43

Table 43: Accuracy of (VGG-Face+VGG-F) + Fisher feature selection with centered subtraction + PCA + MNRML + gaussian SVM blending. KinFacel. PCA dimension optimized for this setting per category (by exhaustive search in range 15-70). Parameters setting:

- SVM's Beta merging

Fisher dimension reduction	PCA dim	fd	fs	md	ms	Mean
0.5	56 50 55 64	0.7388	0.7788	0.8268	0.7888	0.7833
0.4	44 35 49 68	0.7239	0.7788	0.8268	0.7802	0.7774
0.3	69 35 52 69	0.7127	0.7692	0.8346	0.7845	0.7753
0.2	67 45 39 63	0.7239	0.7756	0.8150	0.7672	0.7704
0.1	69 41 30 53	0.7164	0.7436	0.8228	0.7500	0.7582
0.075	47 20 43 28	0.7276	0.7340	0.7913	0.7586	0.7529
0.05	60 45 40 68	0.6903	0.6955	0.7795	0.7414	0.7267

0.025	42 47 49 15	0.6418	0.6667	0.7402	0.7112	0.6900
-------	-------------	--------	--------	--------	--------	--------

Conclusion for tables 38 and 44: Fisher subtraction works better than fisher subtraction with centering.

Finally, in table 44 exhaustive search of PCA and fisher feature selection parameters have been made, trying the specified different values.

Table 44: Best accuracy found for each database. Pipeline: (VGG-Face+VGG-F) + Fisher feature selection with subtraction + PCA + MNRML + gaussian SVM blending. Databases KinFace-I and KinFace-II. Exhaustive search range:

- Fisher dimension search values = [0.5 0.4 0.3 0.2 0.1 0.075 0.05 0.025] (for each feature)
- PCA dimension search: 15:70

Database	Fisher dimension reduction	PCA dim	fd	fs	md	ms	Mean
KinFace-I	[0.025 0.1; 0.1 0.4; 0.5 0.05; 0.1 0.025]	28 70 63 47	0.7948	0.8462	0.8661	0.8707	0.8444
KinFace-II	[0.075 0.4; 0.05 0.05; 0.05 0.05; 0.075 0.075]	60 69 61 58	0.8240	0.8680	0.8900	0.8740	0.8640

Looking at the performance of the pipeline using the features individually (table 1 of the thesis report), it seems the best individual features are VGG-F and HOG. However we have not tried combining only those two. The next table shows the mentioned experiment using the best pipeline configuration and those two features.

Table 45: Accuracy of (VGG-F+HOG) + Fisher feature selection with subtraction + PCA + MNRML + gaussian SVM blending. Databases KinFace-I and KinFace-II:

Database	Fisher dimension reduction	PCA dim	fd	fs	md	ms	Mean
KinFace-I	[0.1 1; 0.4 1; 0.05 1; 0.025 1]	28 70 63 47	0.7052	0.7628	0.8268	0.7629	0.7644
KinFace-II	[0.4 1; 0.05 1; 0.05 1; 0.075 1]	60 69 61 58	0.7580	0.8260	0.8080	0.8280	0.8050