iFAS:
image Fidelity Assessment Software
Guide

B. Ortiz-Jaramillo, L. Platisa and W. Philips

June 27, 2019

Contents

Chapter 1

Getting started

```
iFAS beta-version Edition 2019
Copyleft 2016 Benhur Ortiz-Jaramillo
This program comes with absolutely no warranty
See the GNU General Public License for details
```

1.1 Why do we need iFAS?

Nowadays, multiple image processing areas have shown the necessity of what is called *fidelity assessment*, i.e., objective assessment of differences between a reference and a test image [18, 19, 22]. More specifically, the objective of fidelity assessment is to determine if there are visual differences between a given test image and the corresponding reference image considering certain features and if so to quantify those differences [23,24]. For instance, in the research of color science and imaging technology images are assessed according to the exhibited color differences [18]. This includes image processing tasks such as color correction [14, 16], color quantization [11], color mapping [17], among others. In medical applications, image fidelity is often based on quantifying the visibility between a structure of interest such a vessel and its surrounding anatomical background [19,20]. In such a case, the feature of interest is image contrast and thus the image with the highest fidelity is the one with the lowest contrast difference relative to a standard reference image [19]. In the textile industry, the evaluation is based on the lifetime and appearance retention which is closely related to the surface appearance of the textile material. For example, appearance retention of textile floor coverings is based on measuring fine changes of global texture between the reference (new carpet) and test sample ("used" carpet) [10]. Therefore, the task of fidelity assessment is to measure visual differences between the two images of the textile material under inspection considering texture as the feature of interest [22].

The variety of image domains requiring image fidelity assessment implies the need for application-specific or even application-tailored methodology rather than a one-size-fits-all solution. While many candidate algorithms already exist, testing them and identifying the best ones for a given use case is far from an easy exercise. Thus, it often takes a lot of time and effort to even prepare the test environment for benchmarking, validation and/or developing.

1.2 What is iFAS?

The image Fidelity Assessment Software (iFAS) is a software tool designed to assist researchers, engineers and other users in the process of image fidelity assessment. iFAS provides easy access to a range of state-of-the-art methods as well as intuitive visualizations that aid data analysis. The software is freely available to all for non-commercial use. iFAS includes a range of common mechanisms for image fidelity assessment including computation of fidelity measures on a single pair of images and/or in a full database, visualization of pixel-wise image differences and histogram of the image differences, scatter plots and correlation analysis between human scores and objective measures. Furthermore, iFAS is designed with the purpose of making the interface user friendly even for adding new functions (by modifying or extending the source code). This user interface takes away the interaction of the user with the code, reducing the time spent on the fidelity assessment phase of a merging technology and/or benchmarking of image processing algorithms.

1.3 System requirements

The minimum system requirements to run iFAS are:

- 1000 MHz processor
- 2048 MB RAM
- 400 MB of hard-drive space (or USB stick)
- 3D Acceleration Capable Video card with at least 512 MB

The recommended operating system is GNU Linux Ubuntu 14 or higher [8]. iFAS is developed using Python 2.7 [1]; GTK+ [2]; Scipy 0.17.0 together with its core packages, particularly, NumPy 1.11.0, Matplotlib 1.5.1, pandas 0.17.1, nose 1.3.7, Cython 0.25.1 and Scikits 0.12.3 [7]; PIL 1.1.7 [5]; PyGObject (aka PyGI) 3.20.0 [3]; Pycairo 1.10.0 [4]; PyWavelets 0.5.0 [6]. Therefore, those packages need to be pre-installed before running iFAS. Opencv bindings are also necessary [9]. See the README file included with iFAS software for further information about installation.

1.4 Running iFAS

To run iFAS first make sure that your system has the minimum requirements described in the previous Section and the README file included with iFAS. In the Terminal in Ubuntu Linux,

```
$ cd /folder/to/files/iFAS/
$ python2.7 iFAS.py
```

Afterwards iFAS user interface is open. Figure 1.1 shows a screenshot of iFAS user interface with each of its components. The interface displays seven main components: (1) the main menu, (2) the reference image, (3) the test image, (4) the pixel-wise difference image, (5) the scatter plot of the MOS versus the measure values for a given reference, and (6) the results pane.

The reference image is the reference image selected by the user. The test image is the test image selected after selecting the reference image. The scatter plot of MOS versus the measure values is the

Figure 1.1: iFAS main window screenshot.

space designed for plotting the values of the selected fidelity measures against subjective scores or other fidelity measure values from the current package. In the results pane, the current values of the current session are displayed. The pixel-wise difference image displays the pixel-wise difference of the fidelity measure selected by the user between the reference and test images. In Chapter 2, we describe how to change each of these components.

Chapter 2

iFAS tools

In this chapter we explore every tool available on iFAS. To do so, we use examples step by step showing screenshots of each major step.

2.1 Single Source - Single Sample

This type of analysis only computes the selected fidelity measures between one reference image and one corresponding test image, i.e., the current images displayed on the user interface (images are selected using a file chooser). This analysis is executed as follows on the main menu: File ⇒ New ⇒ Single Source ⇒ Single Sample. After clicking a pop up window with the list of available Python packages is displayed (see Figure 2.1(a)). Here, one and only one package can be selected. By selecting the package a similar pop up window appears displaying the available functions in the package (see Figure 2.1(b)). Here, the user can select one or multiple functions (hold ctrl key + left mouse click) to be computed in the pair of images to be selected. The images are selected right after selecting the methods. The file chooser appears twice, one for the reference and one for the test image (see Figure 2.1(c) and 2.1(d)), respectively. iFAS accept the most common types of image files: BMP, JPEG, JPEG 2000, PNG, PPM, TIFF, XPM [5]. Afterwards, the software starts processing the selected images using the selected methods. The status bar shows the status of the processes in the current pair of images (note that the computational time of these processes depend of the system capabilities).

When the processes have finished, the software displays in the results pane the results of the selected methods and a message identifying that the processes have finished. Additionally, the pixel-wise difference image is updated. These are the details displayed on the results pane:

```
2019-27-06 15:40 14:22:39
Current package:  cd_measures_pack
Reference i03.bmp
Sample i03_07_2.bmp
cd00_deltaE2000 = 6.22467
cd01_SdeltaE2000 = 6.20208
cd02_colordif_mahalanobis = 0.65185
cd03_colorfulness_diff_Gao = 0.01154
cd04_color_SSIM = 0.84766
cd06_colorhist_diff = 0.68961
cd07_weighted_deltaE = 2.20900
Current differnce map:  cd00_deltaE2000
2016-12-05 21:22:39
Process Finished!
```

This is the most basic type of analysis on iFAS used mainly for testing new samples with predefined functions and compare those values to previously computed data.

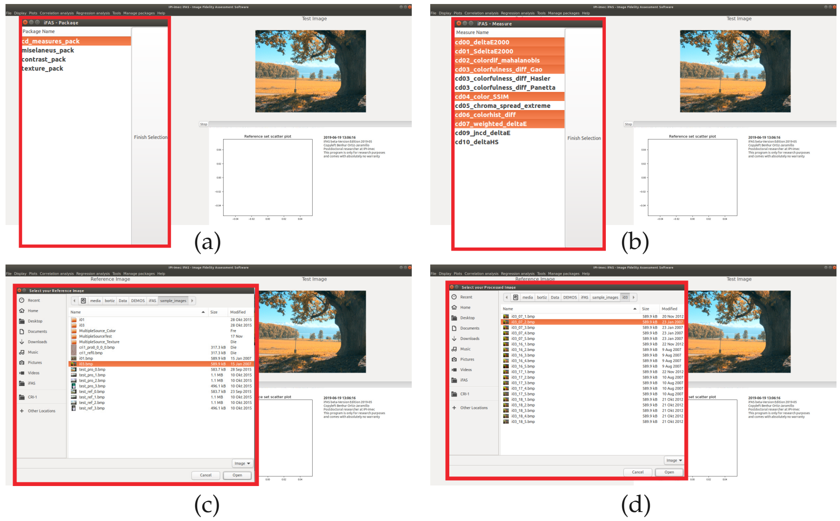


Figure 2.1: All the pop up windows displayed on the Single Source ⇒ Single Sample analysis and Single Source ⇒ Multiple Sample analysis. (a) list of available Python packages, (b) the available functions in the package, (c) the reference and (d) the test image file chooser.

2.2 Single Source - Multiple Sample

This type of analysis computes the fidelity measures selected by the user between one reference image and a number of corresponding test images as specified by the user using a file chooser. The process is very similar to the one explained on Section 2.1, therefore we use the same Figures. This analysis is executed as follows: File ⇒ New ⇒ Single Source ⇒ Multiple Sample. After clicking a pop up window with the list of available Python packages is displayed (see Figure 2.1(a)). Here, one and only one package can be selected. By selecting the package a similar pop up window appears displaying the available functions in the package (see Figure 2.1(b)). Here, the user can select one or multiple functions to be computed in the set of images to be selected. The images are selected right after selecting the methods. The file chooser appears twice, one for the reference and one for the test images (see Figure 2.1(c) and 2.1(d)), respectively. Note that the second file chooser allows now multiple file selection (hold ctrl key + left mouse click). This means that the user can select as many image files as desires. Afterwards, the software starts processing the selected images using the selected methods. The status bar shows the status of the current pair of images (note that the computational time of these processes depend of the system capabilities).

Here we are going to show a couple of options that are very useful for a set of images (reference image plus a number test images). We use as case scenario, a reference with a set of 20 processed images and 5 different color difference measures. In this case, when the process has finished, the software displays the last processed images on the user interface and print the data for all the values. We only display the results for two of the image examples:

```
2019-27-06 15:40:18
Current package:  cd_measures_pack
Reference i01.bmp
Sample i01_18_4.bmp
cd00_deltaE2000 = 11.66716
cd01_SdeltaE2000 = 11.75023
cd02_colordif_mahalanobis = 4.28796
cd04_color_SSIM = 1.71947
cd05_chroma_spread_extreme = 0.21545
Current differnce map:  cd00_deltaE2000
```

```
2019-27-06 15:40:23
Current package:  cd_measures_pack
Sample i01_18_5.bmp
cd00_deltaE2000 = 18.43680
cd01_SdeltaE2000 = 18.08736
cd02_colordif_mahalanobis = 12.94531
cd04_color_SSIM = 1.76997
cd05_chroma_spread_extreme = 0.31109
Current differnce map:  cd00_deltaE2000
Process Finished!
```

Additionally, the pixel-wise difference image is updated.

2.2.1 The scatter plot space

In the scatter plot space is possible to plot all the computed values between two selected methods and/or subjective scores if available and add it by the user (see Section 2.2.4). The scatter plot can be modified as follows Plots ⇒ Mx Vs My. Afterwards two pop up windows (one after the other) with the list of measures (see Figure 2.1(b)) are displayed for selecting the x-axis and y-axis, respectively. Figure 2.2(a) shows the results after executing such a process. The scatter plot shows each point of the computed data with x-markers. This plot can be used for detecting any relationship between the measures for the current reference content at hand. iFAS computes and displays on the results pane the following correlation indexes typically used as performance indicators of fidelity measures: Pearson Coefficient of Correlation (PCC) [12], Spearman Rank Order Coefficient of Correlation (SROCC), [13] and Coefficient of Correlation of Distances (CCD) [25]. Note that these indexes are computed with the values currently displayed on the scatter plot component.

```
2019-06-27 14:32:01
Current x axis:  cd00_deltaE2000
2019-06-27 14:32:01
Current y axis:  cd04_colorSSIM
2019-06-27 14:32:01
```

```
Pearsonr = 0.72673
Spearmanr = 0.64060
Kendalltau = 0.47368
Correlation distance = 0.66875
```

2.2.2 Saving and loading analysis

After computing a set of complex algorithms over a number of images it comes handy to be able to store such data for further analysis. On iFAS current analysis can be stored by using the option File ⇒ Save. The software stores automatically the necessary variables to continue the session another time. iFAS requests the file name and stores it with file extension .iFAS to be recognized as the results from an iFAS analysis. To continue a previous session it is necessary to go to the option File ⇒ Load. The software retrieves automatically the necessary variables to continue the session from an .iFAS file using a file chooser.

2.2.3 Changing the reference, test and difference image

The reference (when there is multiple sources available, see Section 2.3) and test images are changed using Display ⇒ Change display images. Two consecutive pop up windows appear with the list of available

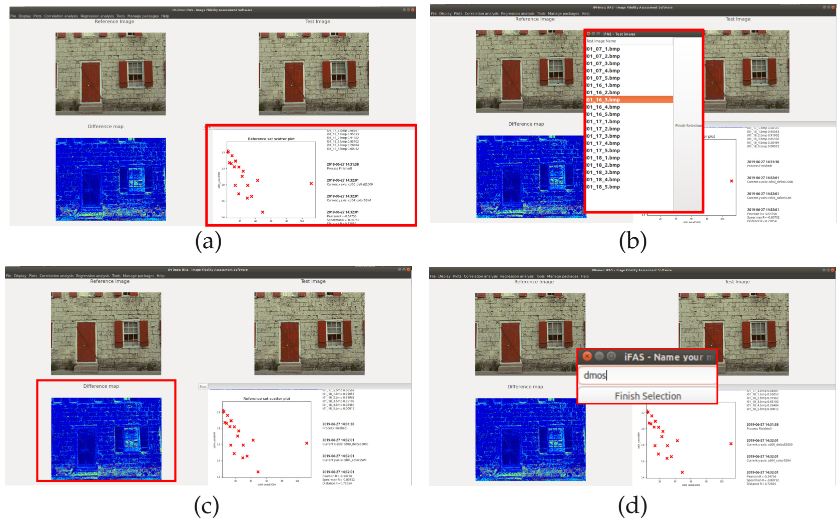


Figure 2.2: Result of the Single Source ⇒ Multiple Sample analysis. (a) scatter plot for the selected methods, (b) reference and test samples, (c) the image difference and (d) naming the new measure values or subjective scores from file.

reference and test images (see Figure 2.2(b)), respectively. Note that the image difference also changes in this case but not the used measure to compute the pixel differences. To change the measure used to compute the image difference is necessary to go to Display ⇒ Difference map. Thereafter, the current image difference can be selected from the displayed list of available fidelity measures (see Figure 2.2(c)).

2.2.4 The MOS scores and other pre-computed methods file

iFAS provides an option to add pre-computed measure values or subjective scores such as MOS or DMOS. This is very useful when there is already pre-computed data from other experiments and/or adding the subjective scores from the database at hand. This can be performed using the option File ⇒ Load pre-computed data. Afterwards, a file chooser appears for the selection of a text file with the values. Note that the user has the responsibility to make sure that the test image corresponds in scene to the reference images and the corresponding values. The text file should be formatted as follows: name reference image, space, name test image, space, the measure value (`file_name_reference file_name_test value`). Every line corresponds to a one and only one test image. For example:

```
ref3.bmp test0.bmp 24.0000000000
ref3.bmp test1.bmp 87.2857142900
```

After selecting the file an input window appears asking for the name of the new added data. The name should be typed by the user and it can be any character combination (see Figure 2.2(d)). However, there is a reserved name by the software which is `human`. `human` name is very important because iFAS uses this variable for many benchamarking analysis using subjective scores (see Section 2.3 for details about this processes). Therefore, the subjective scores in iFAS are recognized with the variable name `human`.

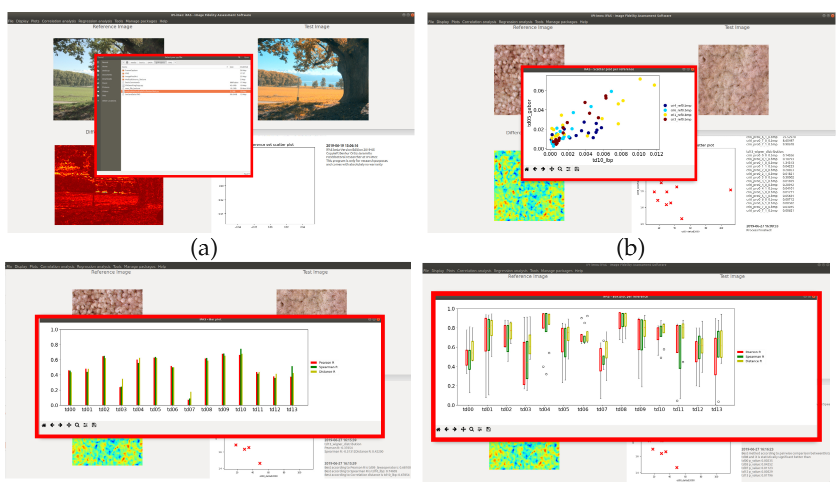
2.3 Multiple Source - Multiple Sample

This type of analysis computes the fidelity measures selected by the user between a number of reference images and a number of test images. This is like executing a Single Source ⇒ Multiple Sample several times using different reference images and their corresponding test samples. Note that, the user has the responsibility to make sure that the test image corresponds in scene to the reference images. This is achieved by preparing a python script with the names of the images with the following format:

```
def myDataBase():
    path2Files = '/path2Files/'
    listReferences = ['reference_name_0', ..., 'reference_name_m']
    dataSet = {'reference_name_0':  ['processed_name_0', ..., 'processed_name_n'],
    ...
    'reference_name_m':  ['processed_name_0', ..., 'processed_name_n']}
    return path2Files, listReferences, dataSet
```

Caution: the test images need to correspond in scene to the reference images in the Python script.

After selecting the file the software performs the operation for every single combination between each reference and its corresponding test images. Note that the computational time of these processes depend of the system capabilities. After finishing the software displays the computed values in the results pane. Afterwards, a range of analysis can be performed with this data. We will describe these analysis in the following paragraphs. To illustrate this type of analysis, we use as example a set of 4 reference images, 7 sample images and 12 different texture measures. We recommend to do this operation by using the provided script 'runDatabase.py'. This script will generate an iFAs file with the necessary data to be loaded in an new session of iFAS to be analyzed. This with the purpose of avoiding screen freezing while computing the fidelity measures.



(a) (b) (c) (d)

Figure 2.3: Result of the Multiple Source - Multiple Sample analysis. (a) Python script selection, (b) scatter plot between fidelity measures for all references, (c) bar plot of the correlation values and (d) box plot of the correlation values per reference.

2.3.1 All data scatter plot

The all data scatter plot is a scatter plot of two selected measures or one measure and subjective scores for all reference samples. In this scatter plot each reference sample is labeled with a different marker color. This option is located on Plots ⇒ Mx Vs My [full data]. Two consecutive pop up windows appear with the list of methods to be plotted. Figure 2.3(b) shows the pop up window with the resulting scatter plot. Note that this figure can be stored in vector format (.eps, .svg) and/or any other image format such as (.jpg, .png, .tif). In Chapter 3 all images are stored in vector format with the purpose of showing the capabilities of iFAS to produce scientific publications type images. For the rest of this Chapter we show only screenshots with the purpose of showing iFAS user interface.

2.3.2 Global correlation analysis

If the user has provided a subjective scores file and it was named `human` (see Section 2.2.4), it is possible to perform a global correlation analysis. The global correlation analysis is located on Correlation analysis ⇒ Global bar plot. Afterwards a pop up window appears with the bar plot of the correlation values between the fidelity measures and the subjective score (`human`). Figure 2.3(c) shows a screenshot of the resulting window. Additionally, the correlation values are printed in the result pane as well as the best performing fidelity measures according to these correlation values. Here, we only show for one of the fidelity measures and for the best performing one.

2019-06-27 17:47:42	2019-06-27 17:47:42
td13_wigner_distribution	
Pearson R: -0.36568	Best according to Pearson R is td09_lawsoperators: 0.68100
Spearman R: -0.50389	Best according to Spearman R is td10_lbp: 0.74605
Distance R: 0.41954	Best according to Correlation distance is td10_lbp: 0.67854

2.3.3 Correlation per reference and pairwise comparisons

A box plot analysis is very useful for identifying how well the measures perform if the content remains the same (reference image). In iFAS this analysis is performed with Correlation analysis ⇒ Per source box plot. Figure 2.3(d) shows the resulting box plot. The correlation values are computed between the fidelity measures and the subjective scores (`human`). The software provides on the results pane a pairwise comparison between the measures with the purpose of identifying if there are statistically significant differences in terms of the performance between the them (cf. Garcia et. al. [15] for details about pairwise comparisons). Here, we show only the example for CCD:

```
2019-06-27 17:57:38
pvalues - Distance R
td00 td01 td02 td03 td04 td05 td06 td07 td08 td09 td10 td11 td12 td13
0.035 1.000 1.000 0.272 1.501 0.353 0.310 0.108 1.647 1.000 0.673 0.499 0.052 0.205
0.035 1.000 1.000 0.272 1.501 0.353 0.310 0.108 1.647 1.000 0.673 0.499 0.052 0.205
0.310 1.728 1.728 1.000 1.924 1.134 1.067 0.612 1.957 1.728 1.501 1.327 0.398 0.866
0.005 0.499 0.499 0.076 1.000 0.108 0.091 0.022 1.200 0.499 0.272 0.176 0.009 0.052
0.237 1.647 1.647 0.866 1.892 1.000 0.933 0.499 1.937 1.647 1.388 1.200 0.310 0.735
0.272 1.690 1.690 0.933 1.909 1.067 1.000 0.554 1.948 1.690 1.446 1.265 0.353 0.800
```

0.612	1.892	1.892	1.388	1.978	1.501	1.446	1.000	1.989	1.892	1.763	1.647	0.735	1.265
0.002	0.353	0.353	0.043	0.800	0.063	0.052	0.011	1.000	0.353	0.176	0.108	0.004	0.028
0.035	1.000	1.000	0.272	1.501	0.353	0.310	0.108	1.647	1.000	0.673	0.499	0.052	0.205
0.091	1.327	1.327	0.499	1.728	0.612	0.554	0.237	1.824	1.327	1.000	0.800	0.128	0.398
0.151	1.501	1.501	0.673	1.824	0.800	0.735	0.353	1.892	1.501	1.200	1.000	0.205	0.554
0.866	1.948	1.948	1.602	1.991	1.690	1.647	1.265	1.996	1.948	1.872	1.795	1.000	1.501
0.398	1.795	1.795	1.134	1.948	1.265	1.200	0.735	1.972	1.795	1.602	1.446	0.499	1.000

Previous values are interpreted as follows: the ith measure is statistically significant better in terms of correlation than the jth measure if the p-value of the ith row and jth column is lower than the significance level, e.g., 0.05. Therefore, considering previous values, iFAS returns the following analysis:

```
2019-06-27 17:57:38
Best method according to pairwise comparison between Distance R
td08 and it is statistically significant better than:
td00 p_value:  0.00235
td03 p_value:  0.04252
td07 p_value:  0.01123
td12 p_value:  0.00406
td13 p_value:  0.02799
```

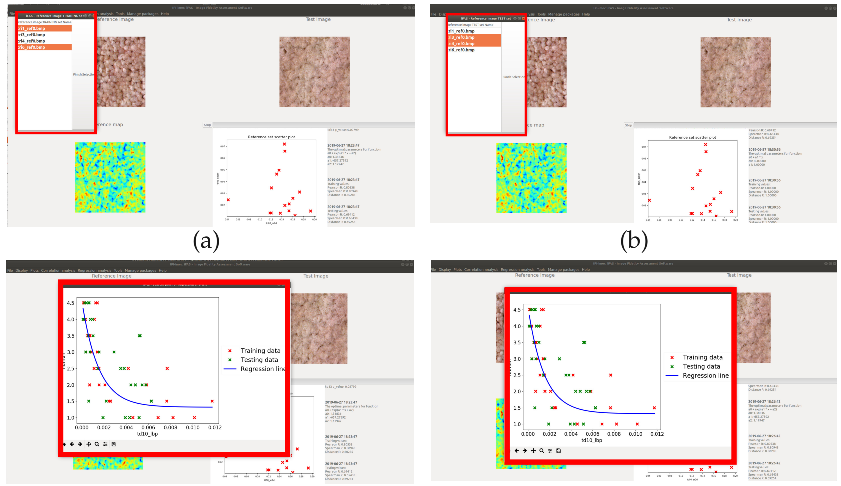
This means that `td08` is the best performing measure but its performance is statistical significant better than `td00`, `td03`, `td07`, `td12` and `td13`.

2.3.4 Non-linear regression

In fidelity assessment is very important to build a model after selecting the appropriated set of measures and test it on unseen data. iFAS simulates such a process by building models using samples provided by the user divided into training and test subsets. The regression analysis is located at Regression analysis $\Rightarrow$ {Linear, Quadratic, Cubic, Exponential, Logistic, Complementary error}. That is, it is possible to fit the data into the six most common mapping functions:

- linear function $y = mx + b$,
- quadratic function $y = ax^2 + bx + c$,
- cubic function $y = ax^3 + bx^2 + cx + d$,
- exponential function $a + \exp(bx + c)$,
- logistic function $a/[1 + \exp(-bx - c)]$ and
- complementary error function $0.5 - 0.5\ \mathrm{erf}[(x - a)/(\sqrt{2}b)]$ [21].

We use the example of Section 2.3.3 to show iFAS regression functions. After selecting the desired type of function, iFAS asks the user to select the set of training and test samples (Figures 2.4(a) and 2.4(b), respectively). This is performed by using two consecutive pop up windows. Here the user can select as many training and test samples as desired. We show on Figures 2.4(c) and 2.4(d) the regression analysis for



(a) (b) (c) (d)

Figure 2.4: Result of the Regression analysis. (a) selection of the training samples, (b) selection of the training samples, (c) screenshot of a cubic model and (d) screenshot of an exponential model.

two references from the example in Section 3.2. The dots are the actual data and the line shows the fitted model using the training data. The parameters of the model are shown in the result pane together with the correlation coefficients for the model. This information comes handy for estimating the performance of the model for external use of the built models. We show as example the information displayed on the result pane for the model on Figure 2.4(d):

```
2019-06-27 18:23:47
The optimal parameters for function
a0 + exp(a1 * x + a2)
a0:  1.31836, a1:  -657.27592, a2:  1.17947
2019-06-27 18:23:47
Training values:
Pearson R: 0.80538
Spearman R: 0.80948
Distance R: 0.80285
2019-06-27 18:23:47
Testing values:
Pearson R: 0.69412
Spearman R: 0.65438
Distance R: 0.69254
```

2.4 Adding new iFAS packages

Packages are added by adding Python scripts. the Python scripts contain functions defined using def and they should return 2 variables. The first one is a difference map of the same size of the input images and the second is the overall fidelity measure. Manage packages ⇒ Add package. Inser the file name of your Python. Afterwards, iFAS will indicate if the software could import the new package successfully.

Chapter 3

Quick examples to use iFAS

In this Chapter we describe and explore use case scenarios with the purpose of showing the advanced tools of iFAS for the analysis of objective fidelity measures. The use cases are the modeling of texture differences on a textile floor covering database, the evaluation of 3 color correction algorithms for a merging a multi-view imaging technology using 2 image color difference measures and the design of a fidelity measure. The set of methods used for the analysis in this Chapter were selected from [18, 21, 22].

3.1 Designing a global contrast measure for images

TODO:

3.2 Textile floor covering assessment using non-linear regression

TODO:

3.3 Improving a multivew imaging system

TODO:

Bibliography

[1] Python software foundation. `https://www.python.org/`, 2016.

[2] The GTK+ project. `https://www.gtk.org/`, 2016.

[3] Python Bindings for GLib/GObject/GIO/GTK+. `https://wiki.gnome.org/Projects/PyGObject`, 2016.

[4] Python Bindings for GLib/GObject/GIO/GTK+. `https://cairographics.org/pycairo/`, 2016.

[5] Python Imaging Library (PIL). `http://www.pythonware.com/products/pil/`, 2016.

[6] PyWavelets - Wavelet Transforms in Python. `https://pywavelets.readthedocs.io/en/latest/`, 2016.

[7] SciPy. `https://www.scipy.org/`, 2016.

[8] Ubuntu. `https://www.ubuntu.com/`, 2016.

[9] OpenCV on Wheels. `https://pypi.org/project/opencv-python/`, 2019.

[10] T. Aibara, T. Mabuchi, and K. Ohue. Automatic evaluation of the appearance of seam puckers on suits. In *Proc. of the SPIE 3652, Machine Vision Applications in Industrial Inspection VII*, pages 1 – 4, 1999.

[11] L. Brun and A. Tremeau. *Digital Color Imaging Handbook*, chapter Color quantization. CRC press, 2002.

[12] P.Y. Chen and P.M. Popovich. *Correlation: Parametric and nonparametric measures*, chapter The Pearson Product-Moment Correlation. Sage Publications, 2002.

[13] P.Y. Chen and P.M. Popovich. *Correlation: Parametric and nonparametric measures*, chapter Special Cases of Pearson's R. Sage Publications, 2002.

[14] S.A. Fezza, M.C. Larabi, and K.M. Faraoun. Feature-based color correction of multiview video for coding and rendering enhancement. *IEEE Transactions on Circuits and Systems for Video Technology*, 24:1486 – 1498, 2014.

[15] S. Garcia, A. Fernandez, J. Luengo, and F. Herrara. Advanced nonparametric tests for multiple comparisons in the design of experiments in computational intelligence and data mining: Experimental analysis of power. *Journal of Information Sciences*, 180:2044 – 2064, 2010.

[16] D.S. Ly, S. Beucher, M. Bilodeau, S. Persa, K.J. Damstra, R. Pot, and J.V. Rooy. Automatic color correction: region-based approach and performance evaluation using full reference metrics. *Journal of Electronic Imaging*, 24:061207 1 – 9, 2015.

[17] J. Morovic. *Color Gamut Mapping*, chapter Desired Color Reproduction Properties and their Evaluation. John Wiley & Sons, Inc., 2008.

[18] B. Ortiz-Jaramillo, A. Kumcu, and W. Philips. Evaluating color difference measures in images. In *Proc. of the Conference on Quality of Multimedia Experience*, pages 1 – 6, 2016.

[19] B. Ortiz-Jaramillo, A. Kumcu, L. Platisa, and W. Philips. Computing contrast ratio in images using local content information. In *Proc. of the Symposium on Signal Processing, Images and Computer Vision*, pages 1 – 6, 2015.

[20] B. Ortiz-Jaramillo, A. Kumcu, L. Platisa, and W. Philips. Computing contrast ratio in medical images using local content information. In *Proc. of the Medical Image Perception Conference*, pages 34 – 34, 2015.

[21] B. Ortiz-Jaramillo, J. Nino-Castaneda, L. Platisa, and W. Philips. Content-aware objective video quality assessment. *Journal of Electronic Imaging*, 25:013011 1 – 16, 2016.

[22] B. Ortiz-Jaramillo, S.A. Orjuela-Vargas, L. Van-Langenhove, C.G. Castellanos-Dominguez, and W. Philips. Reviewing, selecting and evaluating features in distinguishing fine changes of global texture. *Pattern Analysis and Applications*, 17:1 – 15, 2014.

[23] T.N. Pappas, R.J. Safranek, and J. Chen. *Handbook of Image and Video Processing*, chapter Perceptual Criteria for Image Quality Evaluation. Academic Press, 2010.

[24] D.A. Silverstein and J.E. Farrell. The relationship between image fidelity and image quality. In *Proc. of the International Conference on Image Processing*, pages 881 – 884, 1996.

[25] G.J. Székely, M.L. Rizzo, and N.K. Bakirov. Measuring and testing dependence by correlation of distances. *The Annals of Statistics*, 35:2769 – 2794, 2007.