6. Appendix

6.1. User Manual

The Manual section will provide instructions on how to prepare the data/log to be fed to the pretrained model for evaluation and how to export results to .CSV format. In this process Paradigm Geolog is used to export the processed and depth matched logs to a Python readable format, then the provided python scripts will analyse and export results.

6.1.1. Data preparation

the very first step is to depth match the log to ensure the log image is on depth. Next the raw data will need to be processed, normalized and speed corrected to generate Static and Dynamic images of the borehole. For this process we have used Paradigms Geolog.

6.1.1.1. Load data to Project

In the project window select "file importer":

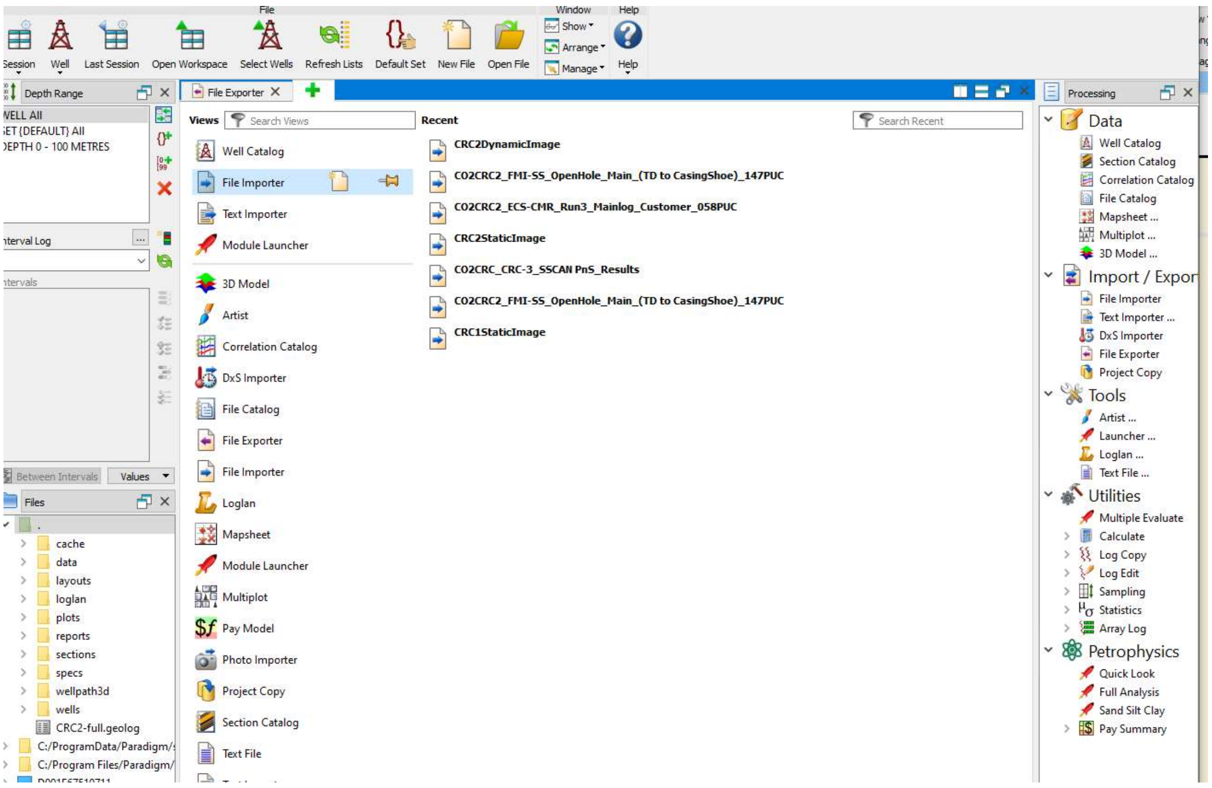


Figure 11 geolog projects window

In the file importer page, select the file you want to import and click on the import button. The well name sets and logs will be imported from the file.

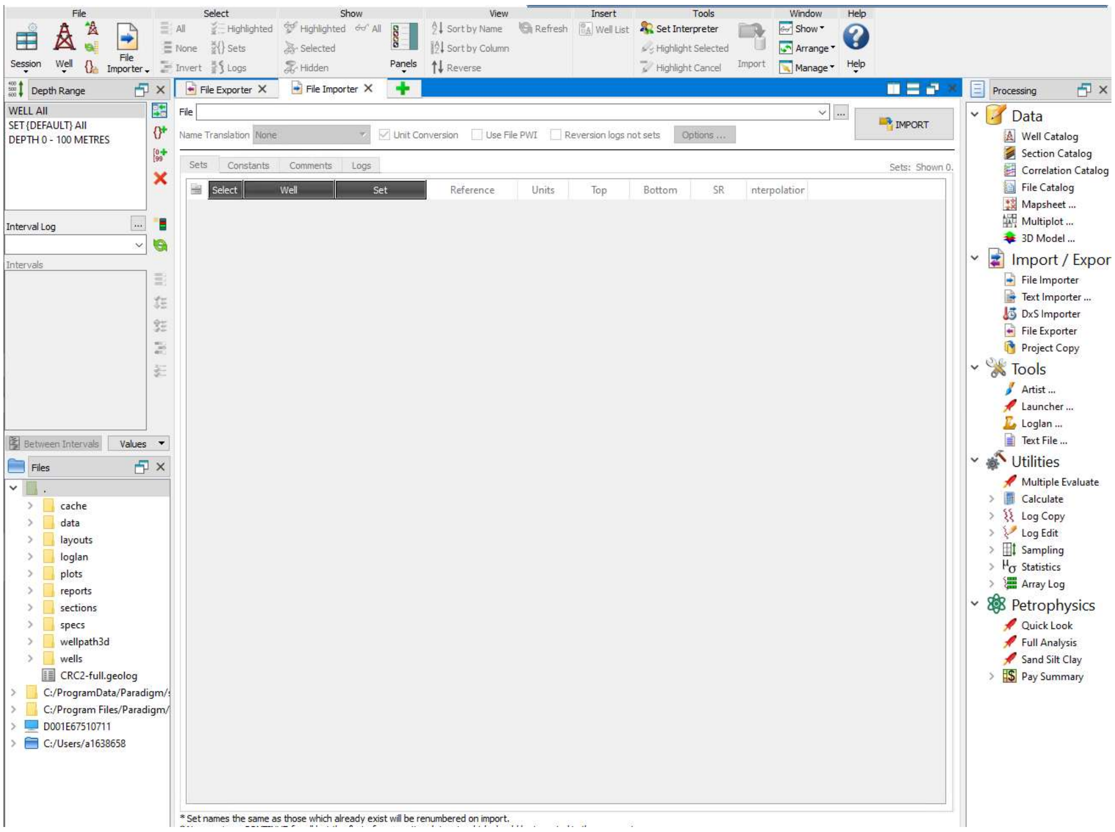
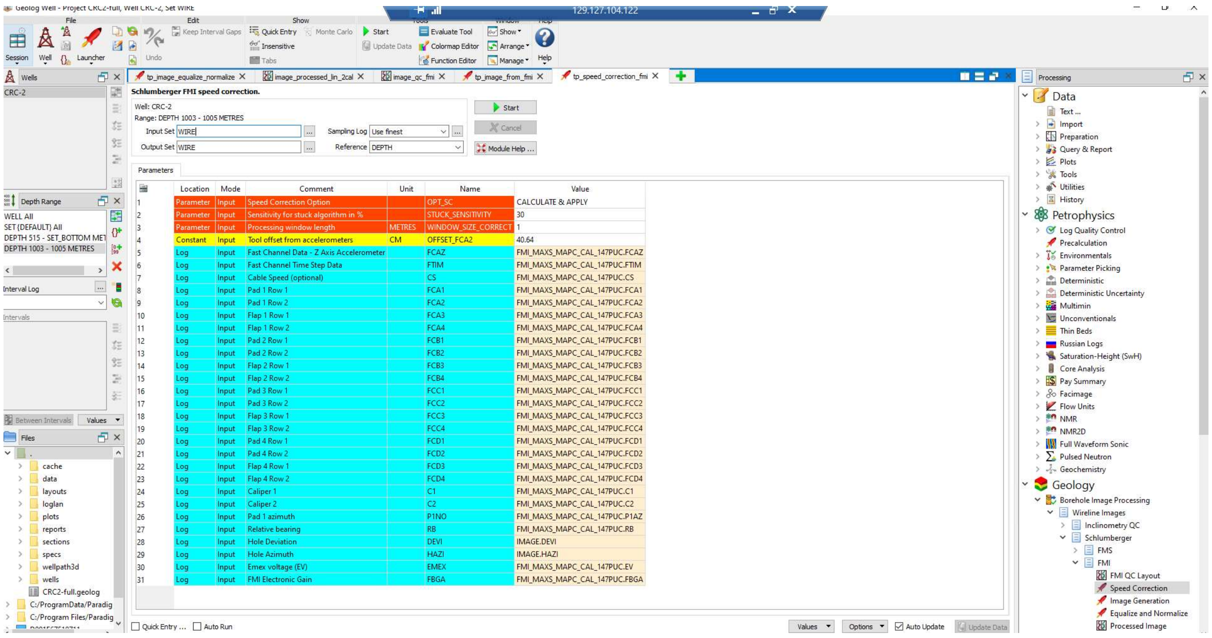


Figure 12 geolog file importer window

6.1.1.2. Speed correction, Normalization and Image generation

The purpose of this step is to derive a processed, normalized and speed corrected Static and Dynamic image of the borehole.
To in the processing tab, go to **Geology -> Borehole image processing -> Wireline images -> Schlumberger -> FMI** And select **Speed correction**.



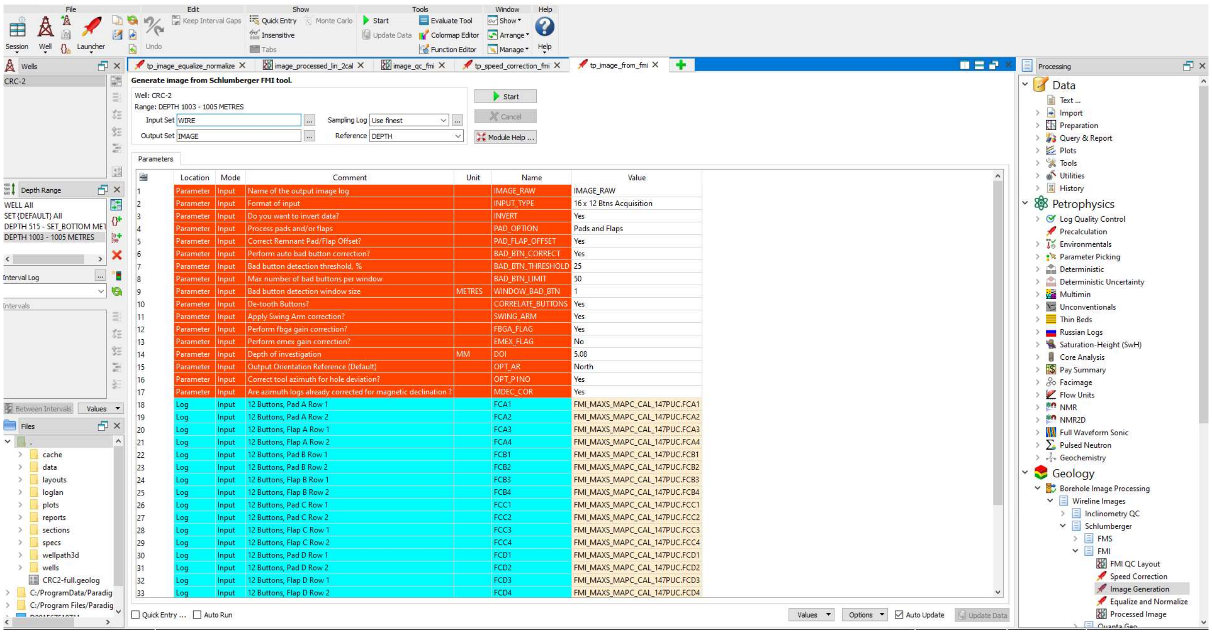
Schlumberger FMI speed correction.

Well: CRC-2
Range: DEPTH 1003 - 1005 METRES
Input Set WIRE
Output Set WIRE
Sampling Log Use finest
Reference DEPTH

	Location	Mode	Comment	Unit	Name	Value
1	Parameter	Input	Speed Correction Option		OPT_SC	CALCULATE & APPLY
2	Parameter	Input	Sensitivity for stuck algorithm in %		STUCK_SENSITIVITY	30
3	Parameter	Input	Processing window length	METRES	WINDOW_SIZE_CORRECT	1
4	Constant	Input	Tool offset from accelerometers	CM	OFFSET_FCA2	40.64
5	Log	Input	Fast Channel Data - Z Axis Accelerometer		FCAZ	FMI_MAXS_MAPC_CAL_147PUC.FCAZ
6	Log	Input	Fast Channel Time Step Data		FTIM	FMI_MAXS_MAPC_CAL_147PUC.FTIM
7	Log	Input	Cable Speed (optional)		CS	FMI_MAXS_MAPC_CAL_147PUC.CS
8	Log	Input	Pad 1 Row 1		FCA1	FMI_MAXS_MAPC_CAL_147PUC.FCA1
9	Log	Input	Pad 1 Row 2		FCA2	FMI_MAXS_MAPC_CAL_147PUC.FCA2
10	Log	Input	Flap 1 Row 1		FCA3	FMI_MAXS_MAPC_CAL_147PUC.FCA3
11	Log	Input	Flap 1 Row 2		FCA4	FMI_MAXS_MAPC_CAL_147PUC.FCA4
12	Log	Input	Pad 2 Row 1		FCB1	FMI_MAXS_MAPC_CAL_147PUC.FCB1
13	Log	Input	Pad 2 Row 2		FCB2	FMI_MAXS_MAPC_CAL_147PUC.FCB2
14	Log	Input	Flap 2 Row 1		FCB3	FMI_MAXS_MAPC_CAL_147PUC.FCB3
15	Log	Input	Flap 2 Row 2		FCB4	FMI_MAXS_MAPC_CAL_147PUC.FCB4
16	Log	Input	Pad 3 Row 1		FCC1	FMI_MAXS_MAPC_CAL_147PUC.FCC1
17	Log	Input	Pad 3 Row 2		FCC2	FMI_MAXS_MAPC_CAL_147PUC.FCC2
18	Log	Input	Flap 3 Row 1		FCC3	FMI_MAXS_MAPC_CAL_147PUC.FCC3
19	Log	Input	Flap 3 Row 2		FCC4	FMI_MAXS_MAPC_CAL_147PUC.FCC4
20	Log	Input	Pad 4 Row 1		FCD1	FMI_MAXS_MAPC_CAL_147PUC.FCD1
21	Log	Input	Pad 4 Row 2		FCD2	FMI_MAXS_MAPC_CAL_147PUC.FCD2
22	Log	Input	Flap 4 Row 1		FCD3	FMI_MAXS_MAPC_CAL_147PUC.FCD3
23	Log	Input	Flap 4 Row 2		FCD4	FMI_MAXS_MAPC_CAL_147PUC.FCD4
24	Log	Input	Caliper 1		C1	FMI_MAXS_MAPC_CAL_147PUC.C1
25	Log	Input	Caliper 2		C2	FMI_MAXS_MAPC_CAL_147PUC.C2
26	Log	Input	Pad 1 azimuth		P1NO	FMI_MAXS_MAPC_CAL_147PUC.P1AZ
27	Log	Input	Relative bearing		RB	FMI_MAXS_MAPC_CAL_147PUC.RB
28	Log	Input	Hole Deviation		DEVI	IMAGE.DEVI
29	Log	Input	Hole Azimuth		HAZI	IMAGE.HAZI
30	Log	Input	Emex voltage (EV)		EMEX	FMI_MAXS_MAPC_CAL_147PUC.EV
31	Log	Input	FMI Electronic Gain		FBGA	FMI_MAXS_MAPC_CAL_147PUC.FBGA

Figure 13 geolog speed correction

In this section all the necessary curves are prepopulated. Make sure you have selected the output set you want the processed data to go to.
Next select **Image Generation** from the same menu.



Generate image from Schlumberger FMI tool.

Well: CRC-2
Range: DEPTH 1003 - 1005 METRES
Input Set: WIRE
Output Set: IMAGE
Sampling Log: Use finest
Reference: DEPTH

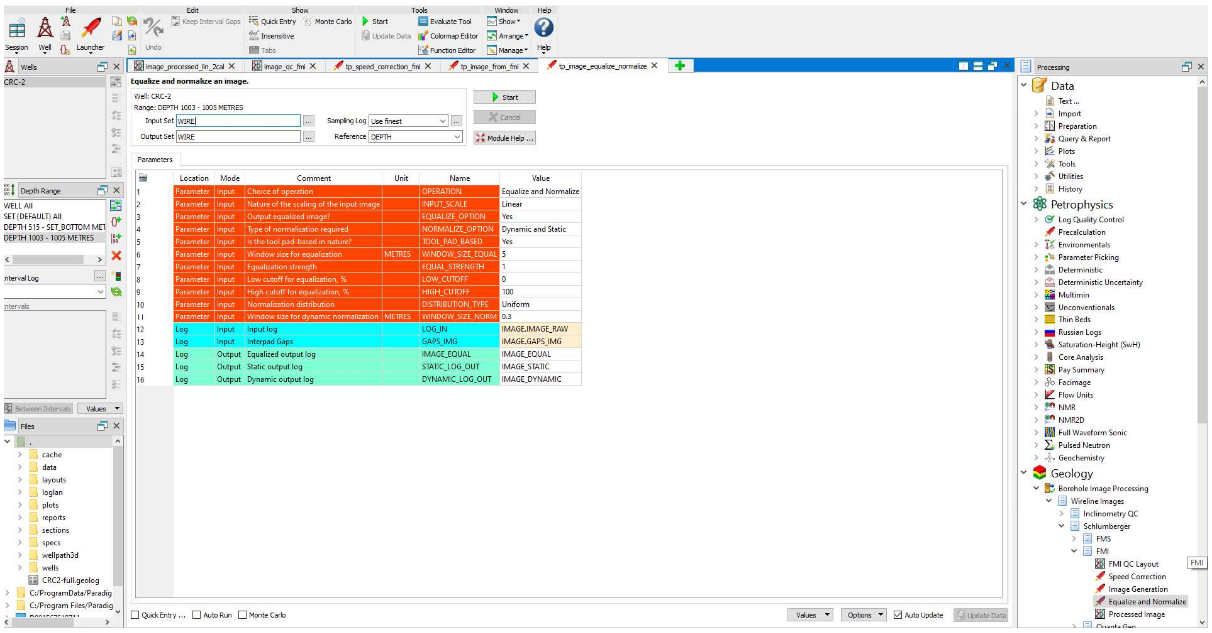
Parameters

	Location	Mode	Comment	Unit	Name	Value
1	Parameter	Input	Name of the output image log		IMAGE_RAW	IMAGE_RAW
2	Parameter	Input	Format of input		INPUT_TYPE	16 x 12 Btns Acquisition
3	Parameter	Input	Do you want to invert data?		INVERT	Yes
4	Parameter	Input	Process pads and/or flaps		PAD_OPTION	Pads and Flaps
5	Parameter	Input	Correct Remnant Pad/Flap Offset?		PAD_FLAP_OFFSET	Yes
6	Parameter	Input	Perform auto bad button correction?		BAD_BTN_CORRECT	Yes
7	Parameter	Input	Bad button detection threshold, %		BAD_BTN_THRESHOLD	25
8	Parameter	Input	Max number of bad buttons per window		BAD_BTN_LIMIT	50
9	Parameter	Input	Bad button detection window size	METRES	WINDOW_BAD_BTN	1
10	Parameter	Input	De-tooth Buttons?		CORRELATE_BUTTONS	Yes
11	Parameter	Input	Apply Swing Arm correction?		SWING_ARM	Yes
12	Parameter	Input	Perform fbga gain correction?		FBGA_FLAG	Yes
13	Parameter	Input	Perform emex gain correction?		EMEX_FLAG	No
14	Parameter	Input	Depth of investigation	MM	DOI	5.08
15	Parameter	Input	Output Orientation Reference (Default)		OPT_AR	North
16	Parameter	Input	Correct tool azimuth for hole deviation?		OPT_P1NO	Yes
17	Parameter	Input	Are azimuth logs already corrected for magnetic declination ?		MDEC_COR	Yes
18	Log	Input	12 Buttons, Pad A Row 1		FCA1	FMI_MAXS_MAPC_CAL_147PUC.FCA1
19	Log	Input	12 Buttons, Pad A Row 2		FCA2	FMI_MAXS_MAPC_CAL_147PUC.FCA2
20	Log	Input	12 Buttons, Flap A Row 1		FCA3	FMI_MAXS_MAPC_CAL_147PUC.FCA3
21	Log	Input	12 Buttons, Flap A Row 2		FCA4	FMI_MAXS_MAPC_CAL_147PUC.FCA4
22	Log	Input	12 Buttons, Pad B Row 1		FCB1	FMI_MAXS_MAPC_CAL_147PUC.FCB1
23	Log	Input	12 Buttons, Pad B Row 2		FCB2	FMI_MAXS_MAPC_CAL_147PUC.FCB2
24	Log	Input	12 Buttons, Flap B Row 1		FCB3	FMI_MAXS_MAPC_CAL_147PUC.FCB3
25	Log	Input	12 Buttons, Flap B Row 2		FCB4	FMI_MAXS_MAPC_CAL_147PUC.FCB4
26	Log	Input	12 Buttons, Pad C Row 1		FCC1	FMI_MAXS_MAPC_CAL_147PUC.FCC1
27	Log	Input	12 Buttons, Pad C Row 2		FCC2	FMI_MAXS_MAPC_CAL_147PUC.FCC2
28	Log	Input	12 Buttons, Flap C Row 1		FCC3	FMI_MAXS_MAPC_CAL_147PUC.FCC3
29	Log	Input	12 Buttons, Flap C Row 2		FCC4	FMI_MAXS_MAPC_CAL_147PUC.FCC4
30	Log	Input	12 Buttons, Pad D Row 1		FCD1	FMI_MAXS_MAPC_CAL_147PUC.FCD1
31	Log	Input	12 Buttons, Pad D Row 2		FCD2	FMI_MAXS_MAPC_CAL_147PUC.FCD2
32	Log	Input	12 Buttons, Flap D Row 1		FCD3	FMI_MAXS_MAPC_CAL_147PUC.FCD3
33	Log	Input	12 Buttons, Flap D Row 2		FCD4	FMI_MAXS_MAPC_CAL_147PUC.FCD4

Figure 14 geolog image generation

Note the name of the generated image in the first row of required data. Make sure you take a note of that.
The second row is to determine the format of the input data. The raw data from the tool is generally 16 X 12 Btns. However, if the data is pre-processed in another software the correct input-type needs to be selected.

Finally select **Equalize and Normalize** menu.



Equalize and normalize an image.

Well: CRC-2
Range: DEPTH 1003 - 1005 METRES
Input Set: WIRE
Output Set: WIRE
Sampling Log: Use finest
Reference: DEPTH

Parameters

	Location	Mode	Comment	Unit	Name	Value
1	Parameter	Input	Choice of operation		OPERATION	Equalize and Normalize
2	Parameter	Input	Nature of the scaling of the input image		INPUT_SCALE	Linear
3	Parameter	Input	Output equalized image?		EQUALIZE_OPTION	Yes
4	Parameter	Input	Type of normalization required		NORMALIZE_OPTION	Dynamic and Static
5	Parameter	Input	Is the tool pad-based in nature?		TOOL_PAD_BASED	Yes
6	Parameter	Input	Window size for equalization	METRES	WINDOW_SIZE_EQUAL	5
7	Parameter	Input	Equalization strength		EQUAL_STRENGTH	1
8	Parameter	Input	Low cutoff for equalization, %		LOW_CUTOFF	0
9	Parameter	Input	High cutoff for equalization, %		HIGH_CUTOFF	100
10	Parameter	Input	Normalization distribution		DISTRIBUTION_TYPE	Uniform
11	Parameter	Input	Window size for dynamic normalization	METRES	WINDOW_SIZE_NORM	0.3
12	Log	Input	Input log		LOG_IN	IMAGE.IMAGE_RAW
13	Log	Input	Interpad Gaps		GAPS_IMG	IMAGE.GAPS_IMG
14	Log	Output	Equalized output log		IMAGE_EQUAL	IMAGE_EQUAL
15	Log	Output	Static output log		STATIC_LOG_OUT	IMAGE_STATIC
16	Log	Output	Dynamic output log		DYNAMIC_LOG_OUT	IMAGE_DYNAMIC

Figure 15 geolog Equalize and normalize window

In this menu make note of the name of the Static and Dynamic images and the set they will be saved to.

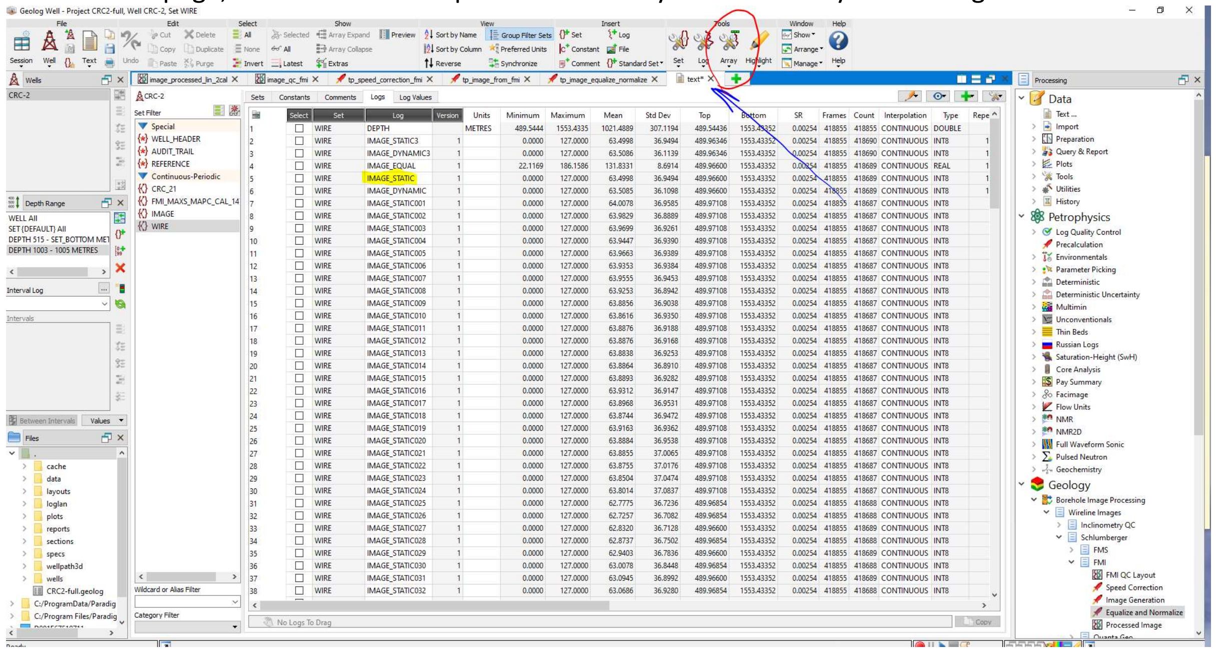
Save wells

Ensure to save the project otherwise the changes will not reflect in other sections of the software.

6.1.1.3. Export images

In this step the processed data and the generated image for which contains the resistivity values recorded by each button the pad is exported to “LAS 2” format, which can be read by the included python scripts.

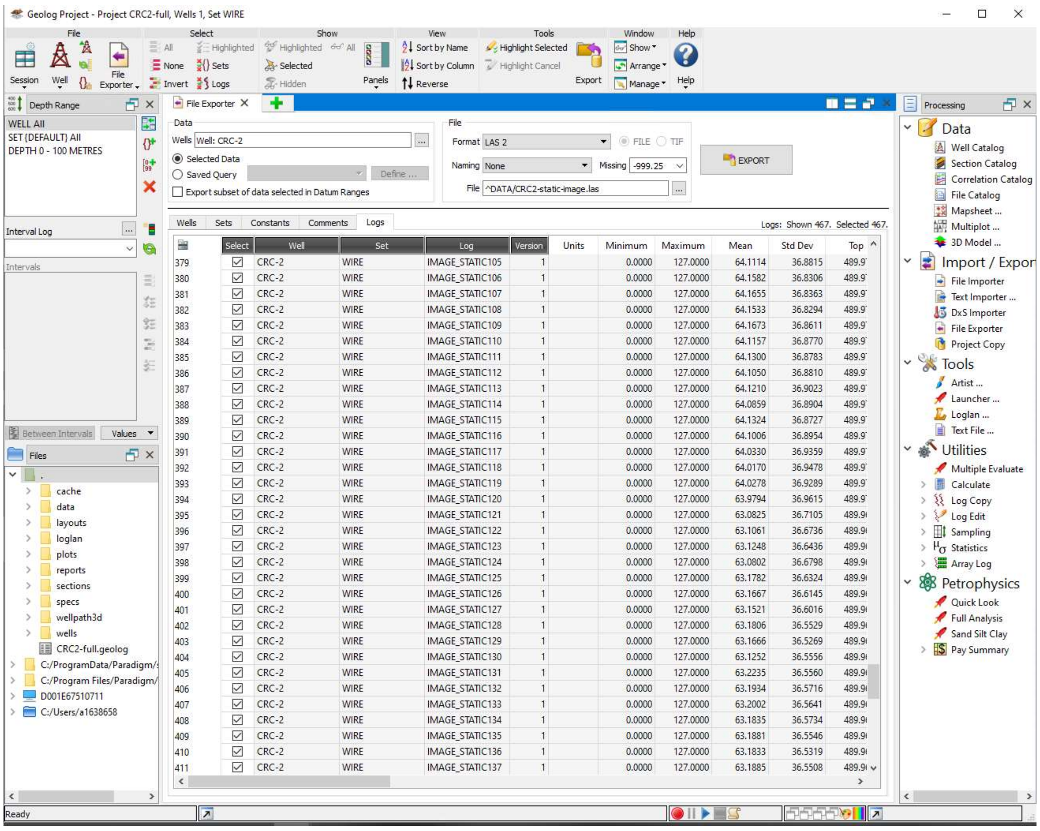
In the “well” page, select the "Text" option and locate your static or dynamic image:



	Select	Set	Log	Version	Units	Minimum	Maximum	Mean	Std Dev	Top	Bottom	SR	Frames	Count	Interpolation	Type	Repe
1	☐	WIRE	DEPTH		METRES	489.5444	1553.4335	1021.4889	307.1194	489.54436	1553.43352	0.00254	418855	418855	CONTINUOUS	DOUBLE	
2	☐	WIRE	IMAGE_STATIC3	1		0.0000	127.0000	63.4998	36.9494	489.96346	1553.43352	0.00254	418855	418690	CONTINUOUS	INT8	1
3	☐	WIRE	IMAGE_DYNAMIC3	1		0.0000	127.0000	63.5086	36.1139	489.96346	1553.43352	0.00254	418855	418690	CONTINUOUS	INT8	1
4	☐	WIRE	IMAGE_EQUAL	1		22.1169	186.1586	131.8331	8.6914	489.96600	1553.43352	0.00254	418855	418689	CONTINUOUS	REAL	1
5	☐	WIRE	IMAGE_STATIC	1		0.0000	127.0000	63.4998	36.9494	489.96600	1553.43352	0.00254	418855	418689	CONTINUOUS	INT8	1
6	☐	WIRE	IMAGE_DYNAMIC	1		0.0000	127.0000	63.5085	36.1098	489.96600	1553.43352	0.00254	418855	418689	CONTINUOUS	INT8	1
7	☐	WIRE	IMAGE_STATIC001	1		0.0000	127.0000	64.0078	36.9585	489.97108	1553.43352	0.00254	418855	418687	CONTINUOUS	INT8	
8	☐	WIRE	IMAGE_STATIC002	1		0.0000	127.0000	63.9829	36.8889	489.97108	1553.43352	0.00254	418855	418687	CONTINUOUS	INT8	
9	☐	WIRE	IMAGE_STATIC003	1		0.0000	127.0000	63.9699	36.9261	489.97108	1553.43352	0.00254	418855	418687	CONTINUOUS	INT8	
10	☐	WIRE	IMAGE_STATIC004	1		0.0000	127.0000	63.9447	36.9390	489.97108	1553.43352	0.00254	418855	418687	CONTINUOUS	INT8	
11	☐	WIRE	IMAGE_STATIC005	1		0.0000	127.0000	63.9663	36.9389	489.97108	1553.43352	0.00254	418855	418687	CONTINUOUS	INT8	
12	☐	WIRE	IMAGE_STATIC006	1		0.0000	127.0000	63.9353	36.9384	489.97108	1553.43352	0.00254	418855	418687	CONTINUOUS	INT8	
13	☐	WIRE	IMAGE_STATIC007	1		0.0000	127.0000	63.9555	36.9453	489.97108	1553.43352	0.00254	418855	418687	CONTINUOUS	INT8	
14	☐	WIRE	IMAGE_STATIC008	1		0.0000	127.0000	63.9253	36.8942	489.97108	1553.43352	0.00254	418855	418687	CONTINUOUS	INT8	
15	☐	WIRE	IMAGE_STATIC009	1		0.0000	127.0000	63.8856	36.9038	489.97108	1553.43352	0.00254	418855	418687	CONTINUOUS	INT8	
16	☐	WIRE	IMAGE_STATIC010	1		0.0000	127.0000	63.8616	36.9350	489.97108	1553.43352	0.00254	418855	418687	CONTINUOUS	INT8	
17	☐	WIRE	IMAGE_STATIC011	1		0.0000	127.0000	63.8876	36.9188	489.97108	1553.43352	0.00254	418855	418687	CONTINUOUS	INT8	
18	☐	WIRE	IMAGE_STATIC012	1		0.0000	127.0000	63.8876	36.9168	489.97108	1553.43352	0.00254	418855	418687	CONTINUOUS	INT8	
19	☐	WIRE	IMAGE_STATIC013	1		0.0000	127.0000	63.8838	36.9253	489.97108	1553.43352	0.00254	418855	418687	CONTINUOUS	INT8	
20	☐	WIRE	IMAGE_STATIC014	1		0.0000	127.0000	63.8864	36.8910	489.97108	1553.43352	0.00254	418855	418687	CONTINUOUS	INT8	
21	☐	WIRE	IMAGE_STATIC015	1		0.0000	127.0000	63.8893	36.9282	489.97108	1553.43352	0.00254	418855	418687	CONTINUOUS	INT8	
22	☐	WIRE	IMAGE_STATIC016	1		0.0000	127.0000	63.9312	36.9147	489.97108	1553.43352	0.00254	418855	418687	CONTINUOUS	INT8	
23	☐	WIRE	IMAGE_STATIC017	1		0.0000	127.0000	63.8968	36.9531	489.97108	1553.43352	0.00254	418855	418687	CONTINUOUS	INT8	
24	☐	WIRE	IMAGE_STATIC018	1		0.0000	127.0000	63.8744	36.9472	489.97108	1553.43352	0.00254	418855	418687	CONTINUOUS	INT8	
25	☐	WIRE	IMAGE_STATIC019	1		0.0000	127.0000	63.9163	36.9362	489.97108	1553.43352	0.00254	418855	418687	CONTINUOUS	INT8	
26	☐	WIRE	IMAGE_STATIC020	1		0.0000	127.0000	63.8884	36.9538	489.97108	1553.43352	0.00254	418855	418687	CONTINUOUS	INT8	
27	☐	WIRE	IMAGE_STATIC021	1		0.0000	127.0000	63.8855	37.0065	489.97108	1553.43352	0.00254	418855	418687	CONTINUOUS	INT8	
28	☐	WIRE	IMAGE_STATIC022	1		0.0000	127.0000	63.8755	37.0176	489.97108	1553.43352	0.00254	418855	418687	CONTINUOUS	INT8	
29	☐	WIRE	IMAGE_STATIC023	1		0.0000	127.0000	63.8504	37.0474	489.97108	1553.43352	0.00254	418855	418687	CONTINUOUS	INT8	
30	☐	WIRE	IMAGE_STATIC024	1		0.0000	127.0000	63.8014	37.0837	489.97108	1553.43352	0.00254	418855	418687	CONTINUOUS	INT8	
31	☐	WIRE	IMAGE_STATIC025	1		0.0000	127.0000	62.7775	36.7236	489.96854	1553.43352	0.00254	418855	418688	CONTINUOUS	INT8	
32	☐	WIRE	IMAGE_STATIC026	1		0.0000	127.0000	62.7257	36.7082	489.96854	1553.43352	0.00254	418855	418688	CONTINUOUS	INT8	
33	☐	WIRE	IMAGE_STATIC027	1		0.0000	127.0000	62.8320	36.7128	489.96600	1553.43352	0.00254	418855	418689	CONTINUOUS	INT8	
34	☐	WIRE	IMAGE_STATIC028	1		0.0000	127.0000	62.8737	36.7502	489.96854	1553.43352	0.00254	418855	418688	CONTINUOUS	INT8	
35	☐	WIRE	IMAGE_STATIC029	1		0.0000	127.0000	62.9403	36.7836	489.96600	1553.43352	0.00254	418855	418689	CONTINUOUS	INT8	
36	☐	WIRE	IMAGE_STATIC030	1		0.0000	127.0000	63.0078	36.8448	489.96854	1553.43352	0.00254	418855	418688	CONTINUOUS	INT8	
37	☐	WIRE	IMAGE_STATIC031	1		0.0000	127.0000	63.0945	36.8992	489.96600	1553.43352	0.00254	418855	418689	CONTINUOUS	INT8	
38	☐	WIRE	IMAGE_STATIC032	1		0.0000	127.0000	63.0686	36.9280	489.96854	1553.43352	0.00254	418855	418688	CONTINUOUS	INT8	

Figure 16 geolog array to scalar function

Select the image and in the "Array" menu select "**Array to scaler**". This will convert the processed/corrected image to scalar values.
After saving these changes they will appear in the project page and can be exported.
In the Project page select "file exporter", select all the scaler curves generated in the previous step and export to "LAS 2", ensure to include the depth curve for reference.



Geolog Project - Project CRC2-full, Wells 1, Set WIRE

WELL All
SET {DEFAULT} All
DEPTH 0 - 100 METRES

Files: cache, data, layouts, loglan, plots, reports, sections, specs, wellpath3d, wells, CRC2-full.geolog, C:/ProgramData/Paradigm/, C:/Program Files/Paradigm/, D001E675107I1, C:/Users/a1638658

Processing: Data (Well Catalog, Section Catalog, Correlation Catalog, File Catalog, Mapsheet ..., Multiplot ..., 3D Model ...); Import / Export (File Importer, Text Importer ..., DxS Importer, File Exporter, Project Copy); Tools (Artist ..., Launcher ..., Loglan ..., Text File ...); Utilities (Multiple Evaluate, Calculate, Log Copy, Log Edit, Sampling, Statistics, Array Log); Petrophysics (Quick Look, Full Analysis, Sand Silt Clay, Pay Summary)

File Exporter

Data
Wells: Well: CRC-2
[✓ Selected Data] Saved Query
Export subset of data selected in Datum Ranges: ☐

File
Format: LAS 2 [✓ FILE] TIF
Naming: None Missing: -999.25
File: ^DATA/CRC2-static-image.las

EXPORT

Wells | Sets | Constants | Comments | Logs

Logs: Shown 467. Selected 467.

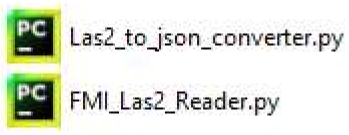
	Select	Well	Set	Log	Version	Units	Minimum	Maximum	Mean	Std Dev	Top
379	✓	CRC-2	WIRE	IMAGE_STATIC105	1		0.0000	127.0000	64.1114	36.8815	489.9
380	✓	CRC-2	WIRE	IMAGE_STATIC106	1		0.0000	127.0000	64.1582	36.8306	489.9
381	✓	CRC-2	WIRE	IMAGE_STATIC107	1		0.0000	127.0000	64.1655	36.8363	489.9
382	✓	CRC-2	WIRE	IMAGE_STATIC108	1		0.0000	127.0000	64.1533	36.8294	489.9
383	✓	CRC-2	WIRE	IMAGE_STATIC109	1		0.0000	127.0000	64.1673	36.8611	489.9
384	✓	CRC-2	WIRE	IMAGE_STATIC110	1		0.0000	127.0000	64.1157	36.8770	489.9
385	✓	CRC-2	WIRE	IMAGE_STATIC111	1		0.0000	127.0000	64.1300	36.8783	489.9
386	✓	CRC-2	WIRE	IMAGE_STATIC112	1		0.0000	127.0000	64.1050	36.8810	489.9
387	✓	CRC-2	WIRE	IMAGE_STATIC113	1		0.0000	127.0000	64.1210	36.9023	489.9
388	✓	CRC-2	WIRE	IMAGE_STATIC114	1		0.0000	127.0000	64.0859	36.8904	489.9
389	✓	CRC-2	WIRE	IMAGE_STATIC115	1		0.0000	127.0000	64.1324	36.8727	489.9
390	✓	CRC-2	WIRE	IMAGE_STATIC116	1		0.0000	127.0000	64.1006	36.8954	489.9
391	✓	CRC-2	WIRE	IMAGE_STATIC117	1		0.0000	127.0000	64.0330	36.9359	489.9
392	✓	CRC-2	WIRE	IMAGE_STATIC118	1		0.0000	127.0000	64.0170	36.9478	489.9
393	✓	CRC-2	WIRE	IMAGE_STATIC119	1		0.0000	127.0000	64.0278	36.9289	489.9
394	✓	CRC-2	WIRE	IMAGE_STATIC120	1		0.0000	127.0000	63.9794	36.9615	489.9
395	✓	CRC-2	WIRE	IMAGE_STATIC121	1		0.0000	127.0000	63.0825	36.7105	489.9
396	✓	CRC-2	WIRE	IMAGE_STATIC122	1		0.0000	127.0000	63.1061	36.6736	489.9
397	✓	CRC-2	WIRE	IMAGE_STATIC123	1		0.0000	127.0000	63.1248	36.6436	489.9
398	✓	CRC-2	WIRE	IMAGE_STATIC124	1		0.0000	127.0000	63.0802	36.6798	489.9
399	✓	CRC-2	WIRE	IMAGE_STATIC125	1		0.0000	127.0000	63.1782	36.6324	489.9
400	✓	CRC-2	WIRE	IMAGE_STATIC126	1		0.0000	127.0000	63.1667	36.6145	489.9
401	✓	CRC-2	WIRE	IMAGE_STATIC127	1		0.0000	127.0000	63.1521	36.6016	489.9
402	✓	CRC-2	WIRE	IMAGE_STATIC128	1		0.0000	127.0000	63.1806	36.5529	489.9
403	✓	CRC-2	WIRE	IMAGE_STATIC129	1		0.0000	127.0000	63.1666	36.5269	489.9
404	✓	CRC-2	WIRE	IMAGE_STATIC130	1		0.0000	127.0000	63.1252	36.5556	489.9
405	✓	CRC-2	WIRE	IMAGE_STATIC131	1		0.0000	127.0000	63.2235	36.5560	489.9
406	✓	CRC-2	WIRE	IMAGE_STATIC132	1		0.0000	127.0000	63.1934	36.5716	489.9
407	✓	CRC-2	WIRE	IMAGE_STATIC133	1		0.0000	127.0000	63.2002	36.5641	489.9
408	✓	CRC-2	WIRE	IMAGE_STATIC134	1		0.0000	127.0000	63.1835	36.5734	489.9
409	✓	CRC-2	WIRE	IMAGE_STATIC135	1		0.0000	127.0000	63.1881	36.5546	489.9
410	✓	CRC-2	WIRE	IMAGE_STATIC136	1		0.0000	127.0000	63.1833	36.5319	489.9
411	✓	CRC-2	WIRE	IMAGE_STATIC137	1		0.0000	127.0000	63.1885	36.5508	489.9

Ready

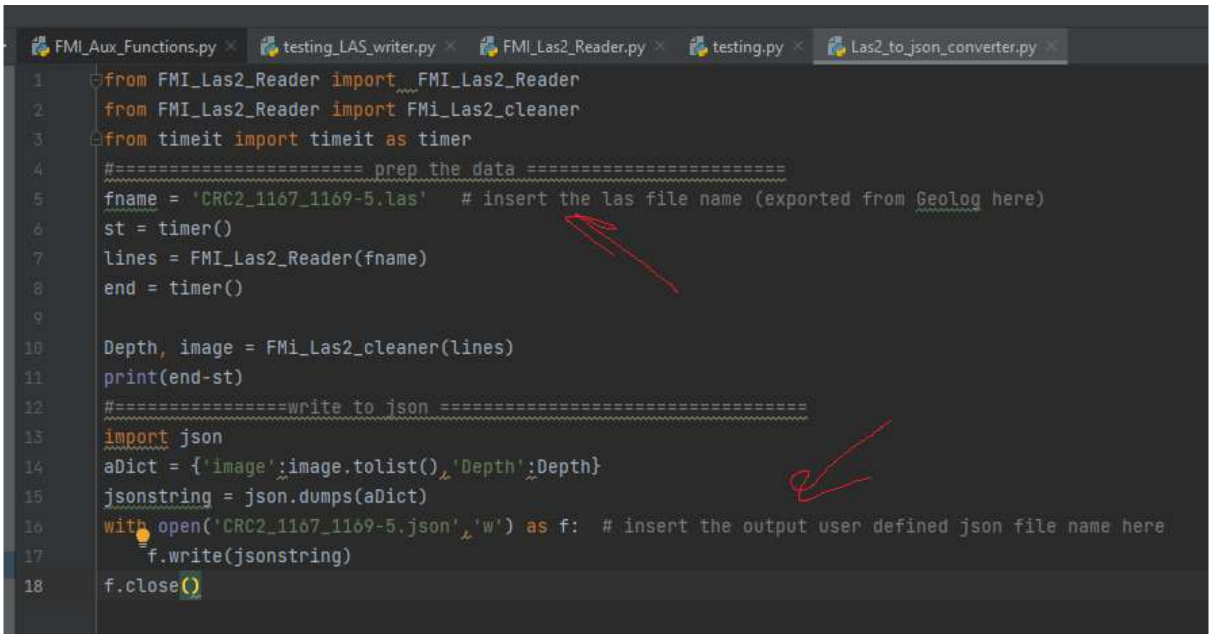
Figure 17 geolog file exported window

6.1.1.4. Convert “Las 2” to python readable format

The following .py files will convert the Las 2 export to json format to be processed by the pretrained network. Both of the below files will need to be in the same directory or python project.



The main file is “Las2_to_json_converter.py”, this script will invoke any required function from “FMI_Las2_Reader.py” automatically. User is required to manually input the Las 2 file name and the output json file in line 5 and 16 respectively.



```python
from FMI_Las2_Reader import  FMI_Las2_Reader
from FMI_Las2_Reader import FMi_Las2_cleaner
from timeit import timeit as timer
#======================= prep the data ========================
fname = 'CRC2_1167_1169-5.las'   # insert the las file name (exported from Geolog here)
st = timer()
lines = FMI_Las2_Reader(fname)
end = timer()

Depth, image = FMi_Las2_cleaner(lines)
print(end-st)
#================write to json =================================
import json
aDict = {'image':image.tolist(),'Depth':Depth}
jsonstring = json.dumps(aDict)
with open('CRC2_1167_1169-5.json','w') as f:  # insert the output user defined json file name here
    f.write(jsonstring)
f.close()
```

Figure 18 location to insert file addresses in the python code

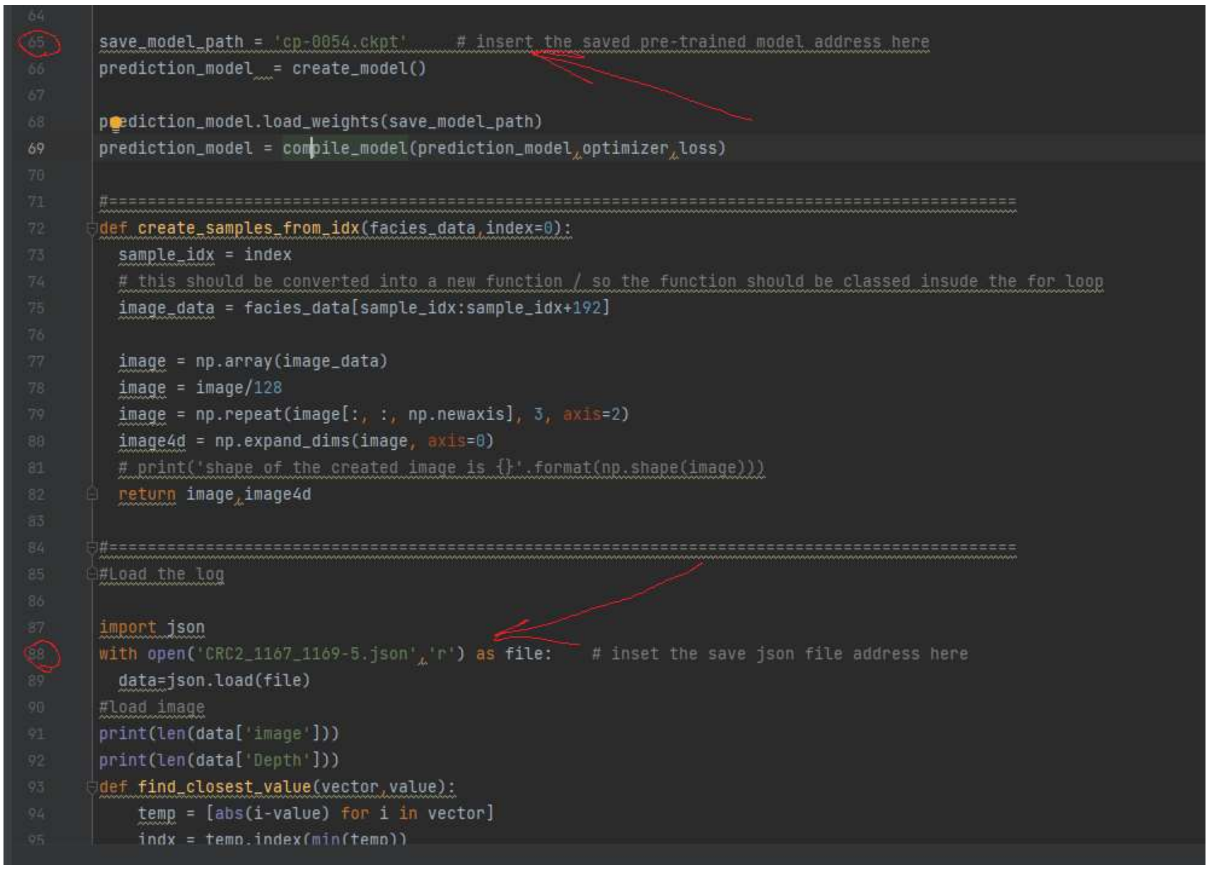
6.1.2. Facies detection

For the facies detection step the following .py script files are required:

final_inference.py

FMI_Aux_Functions.py

The file “final_inference.py” will automatically, load the saved network, and has all the required function built in. the user only needs to ensure that the address where the save network is saved is inserted in line 65 of the code and the address where the raw data in json format is saved is inserted in line 88. (see image below)



```
save_model_path = 'cp-0054.ckpt'     # insert the saved pre-trained model address here
prediction_model_ = create_model()

prediction_model.load_weights(save_model_path)
prediction_model = compile_model(prediction_model,optimizer,loss)

#==============================================================================================
def create_samples_from_idx(facies_data,index=0):
  sample_idx = index
  # this should be converted into a new function / so the function should be classed insude the for loop
  image_data = facies_data[sample_idx:sample_idx+192]

  image = np.array(image_data)
  image = image/128
  image = np.repeat(image[:, :, np.newaxis], 3, axis=2)
  image4d = np.expand_dims(image, axis=0)
  # print('shape of the created image is {}'.format(np.shape(image)))
  return image,image4d

#==============================================================================================
#Load the log

import json
with open('CRC2_1167_1169-5.json','r') as file:    # inset the save json file address here
  data=json.load(file)
#load image
print(len(data['image']))
print(len(data['Depth']))
def find_closest_value(vector,value):
    temp = [abs(i-value) for i in vector]
    indx = temp.index(min(temp))
```

Figure 19 location to insert file addresses in the python code

This file will load the image form the json format. Run a 192 X 192 window over the image and label all depths. Then the results will be exported in .CSV format the same python project.

6.1.3. Example

As an example the scripts were used to export and identify the facies in the interval 1167.0 m – 1169.5 m of well CRC-2 the results are delivered in file format.

6.2. Delivered data

As deliverables for this project the following were submitted:

1. Project report and Manual in .PDF format
2. Project presentation to CO2CRC in .PPX format
3. Four .py python scripts
4. Pre-trained, customized inception V3 network
5. Example files.