National Institute of Technology Karnataka, Surathkal
Department of Electronics and Communication Engineering

Final Report: IP mini project

Guide: Prof. Deepu Vijayasenan
Janavi N :-181EC127
Chandravaran K:-181EC156

Title of the Project

Industry Welding defects in radiographic images

Table of Contents

Objectives

- **The main objective of this project is to take the X-rays from the industry and determine the type of defect.**
- **We are going to deal with 30 types of defects.(roughly)**
- **Few of them are:**

1) **Toe Crack**
2) **Transverse Crack**
3) **Crater Crack**
4) **Root Crack**
5) **Weld Spatter**
6) **Undercut**
7) **Excess Cap**
8) **Concave Crack**

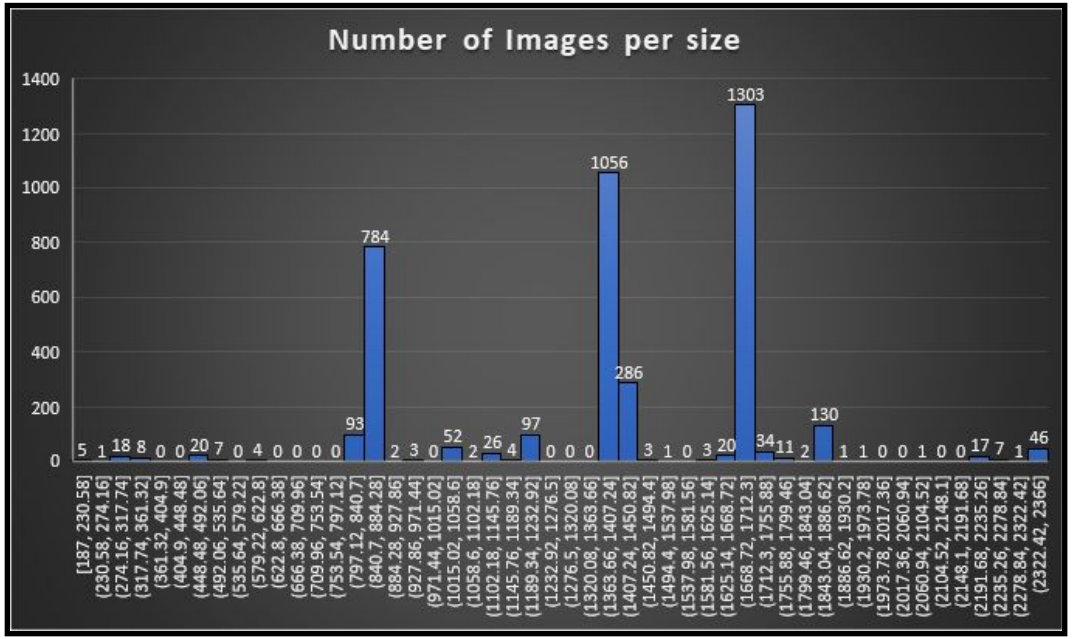
Technology Stack (needed)

Tech Stack used(to be used) in this project :

1. OpenCV (module)
2. Python (language)
3. Jupyter Notebook(ide)
4. Keras,Tensorflow
5. ResNet-50 model architecture
6. Inception_V3
7. Tensorflow hub
8. Data set received from the Industry through our guide deepu

Methodology

Results



Task 1

Analysing the data set for number of vertically stacked images.

As the number of X-rays that are stacked in each image is different.

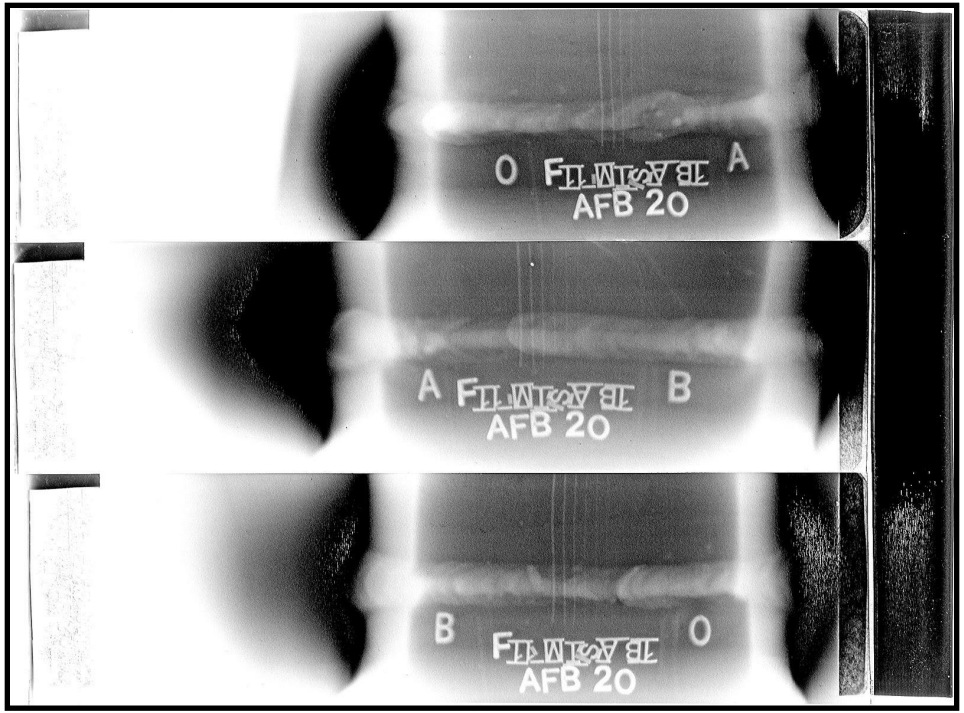
But what we observed is that the based on the bin analysis shown on the left there were 4 classes, single staked, doubly, triple and quadruple stacked.

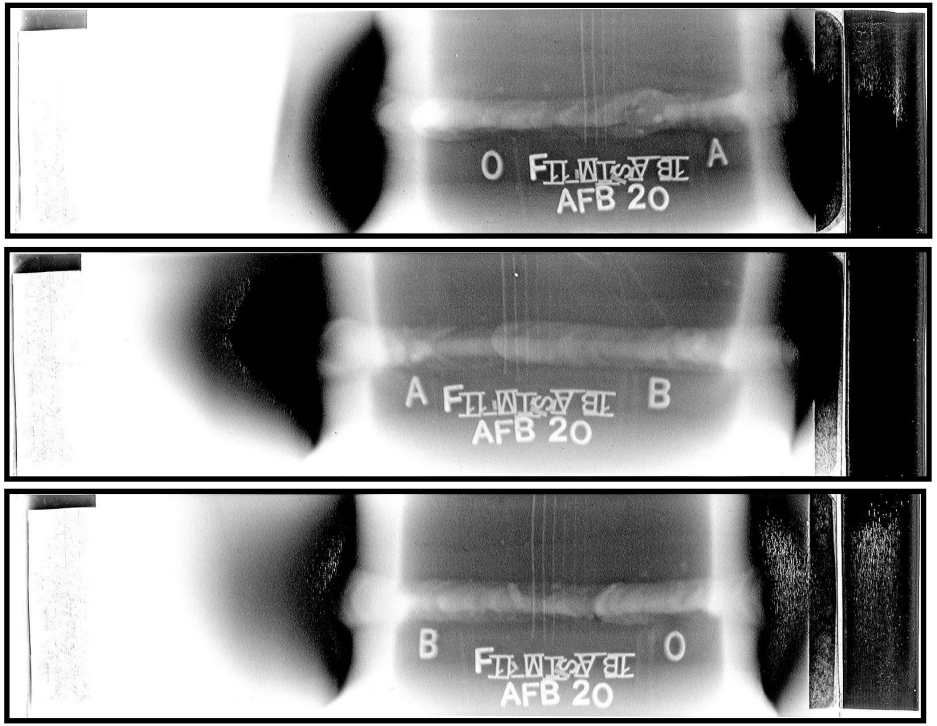
Methodology

Task 2

- Splitting the images. And taking all the lines from above that horizontal line we were able to split.
- We initially tried to determine the horizontal line using hough transform and morphology but It worked only as guidance to see the location of stacked images
- Then we noticed a symmetry among the images that each were split equally among each other so we developed a function that can be seen in the next slide how we cut the images in a systematic way.

Spliting Images





Algo for task-2

Task 2

- There were 2 different methods to finding horizontal lines the images
- The first one was using edge detection and hough transform
- The second one was using morphology, This gave us a better result
- For splitting the images each bin had a fixed number of images that were stacked so to the function that we developed we passed the number of x-rays present and the bin limits so we find images only present in the bin and we cut the images so our program was flexible.

Methodology

Task 3

- Now after the data is split we need to start classifying the images. But the problem at hand is that we don't have labeled data, so we decided to tackle this problem by using the 30 images that were given to us, as sample defects as labels along with the 30 non defective images and the feature vectors were found for all the images
- But before we could use the images we noticed that the images needed to be resized as the model takes in around 299*299. We initially just resized the images directly from the original size to the target size. Results in next slide.

Getting proper size

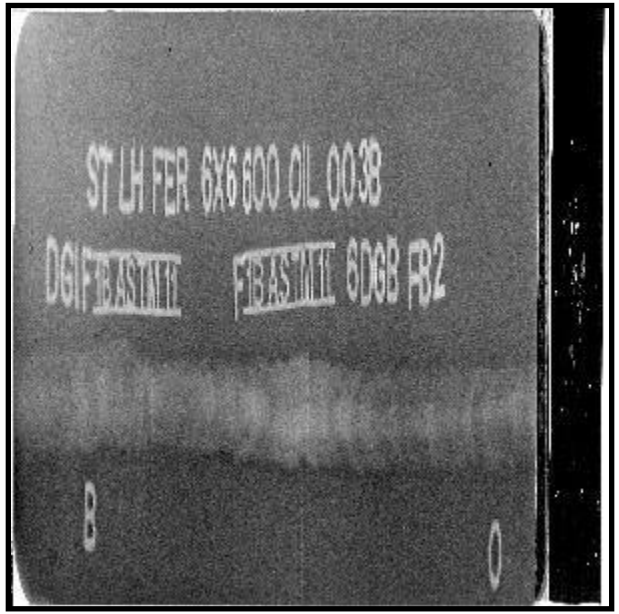


Fig: Resized from original image

Fig: Cut from original image

Results

- So what observed was that the most information required for us was in the middle region of the image, so we decided to cut the image to about three different sizes.
- After which we resized to the required size of 299*299 and 299*500.
- This way the most amount of data can be in the image as well as we can run the model.
- Now the data is ready for feature extraction.

Methodology

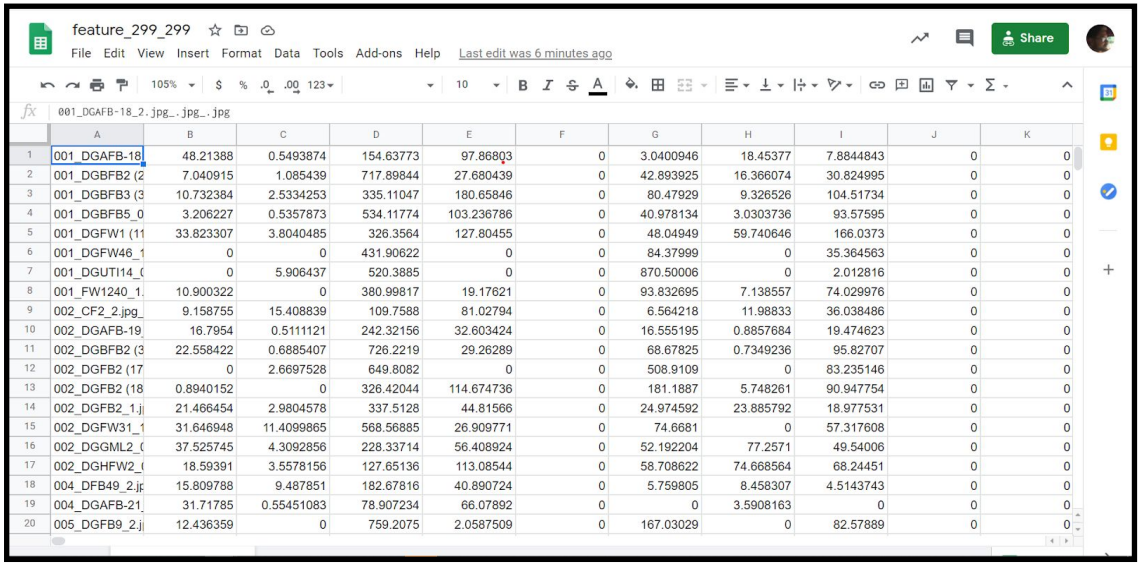
Task-4

- We do not have labeled data which makes it hard to progress with the project, we will be getting the labeled data but that may take some time, so we have decided to use transfer learning and determine the features from a pretrained model, upon which we will run 100 test images and find the mean least square error from each label image and the one with the least distance will be considered as the classified one

- 2 models were obtained from the Tensorflow hub which are called resnet_v2_50 and inception_v3 which has been trained on imagenet data set.

- both give us a feature vector of size 2048
  Links :
- https://tfhub.dev/google/imagenet/resnet_v2_50/feature_vector/4
- https://tfhub.dev/google/imagenet/inception_v3/feature_vector/4

Feature vector of Resnet50



feature_299_299

File Edit View Insert Format Data Tools Add-ons Help Last edit was 6 minutes ago

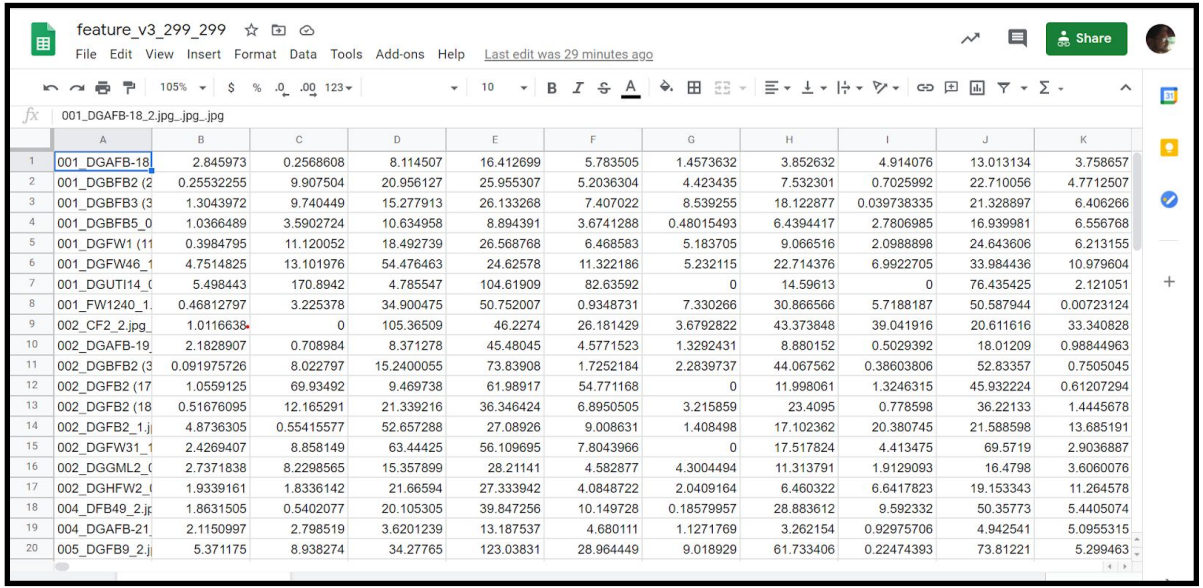
Share

001_DGAFB-18_2.jpg_.jpg_.jpg

	A	B	C	D	E	F	G	H	I	J	K
1	001_DGAFB-18	48.21388	0.5493874	154.63773	97.86803	0	3.0400946	18.45377	7.8844843	0	0
2	001_DGBFB2 (2	7.040915	1.085439	717.89844	27.680439	0	42.893925	16.366074	30.824995	0	0
3	001_DGBFB3 (3	10.732384	2.5334253	335.11047	180.65846	0	80.47929	9.326526	104.51734	0	0
4	001_DGBFB5_0	3.206227	0.5357873	534.11774	103.236786	0	40.978134	3.0303736	93.57595	0	0
5	001_DGFW1 (11	33.823307	3.8040485	326.3564	127.80455	0	48.04949	59.740646	166.0373	0	0
6	001_DGFW46_1	0	0	431.90622	0	0	84.37999	0	35.364563	0	0
7	001_DGUTI14_(	0	5.906437	520.3885	0	0	870.50006	0	2.012816	0	0
8	001_FW1240_1.	10.900322	0	380.99817	19.17621	0	93.832695	7.138557	74.029976	0	0
9	002_CF2_2.jpg_	9.158755	15.408839	109.7588	81.02794	0	6.564218	11.98833	36.038486	0	0
10	002_DGAFB-19_	16.7954	0.5111121	242.32156	32.603424	0	16.555195	0.8857684	19.474623	0	0
11	002_DGBFB2 (3	22.558422	0.6885407	726.2219	29.26289	0	68.67825	0.7349236	95.82707	0	0
12	002_DGFB2 (17	0	2.6697528	649.8082	0	0	508.9109	0	83.235146	0	0
13	002_DGFB2 (18	0.8940152	0	326.42044	114.674736	0	181.1887	5.748261	90.947754	0	0
14	002_DGFB2_1.j	21.466454	2.9804578	337.5128	44.81566	0	24.974592	23.885792	18.977531	0	0
15	002_DGFW31_1	31.646948	11.4099865	568.56885	26.909771	0	74.6681	0	57.317608	0	0
16	002_DGGML2_(	37.525745	4.3092856	228.33714	56.408924	0	52.192204	77.2571	49.54006	0	0
17	002_DGHFW2_(	18.59391	3.5578156	127.65136	113.08544	0	58.708622	74.668564	68.24451	0	0
18	004_DFB49_2.j	15.809788	9.487851	182.67816	40.890724	0	5.759805	8.458307	4.5143743	0	0
19	004_DGAFB-21_	31.71785	0.55451083	78.907234	66.07892	0	0	3.5908163	0	0	0
20	005_DGFB9_2.j	12.436359	0	759.2075	2.0587509	0	167.03029	0	82.57889	0	0

- On the left are the first 10 Feature vectors of the first 20 sample images we took.
- These are the feature vectors obtained for images of size 299*299 we have not displayed the feature vectors for the other set of images as it will be redundant, but have been included in the report.
- From quick scan through the vectors it was observed that Resnet50 had lots of columns with 0 values, this makes me believe that that it was successful in determining the features properly

Feature vector of InceptionV3



feature_v3_299_299

File Edit View Insert Format Data Tools Add-ons Help Last edit was 29 minutes ago

Share

001_DGAFB-18_2.jpg_.jpg_.jpg

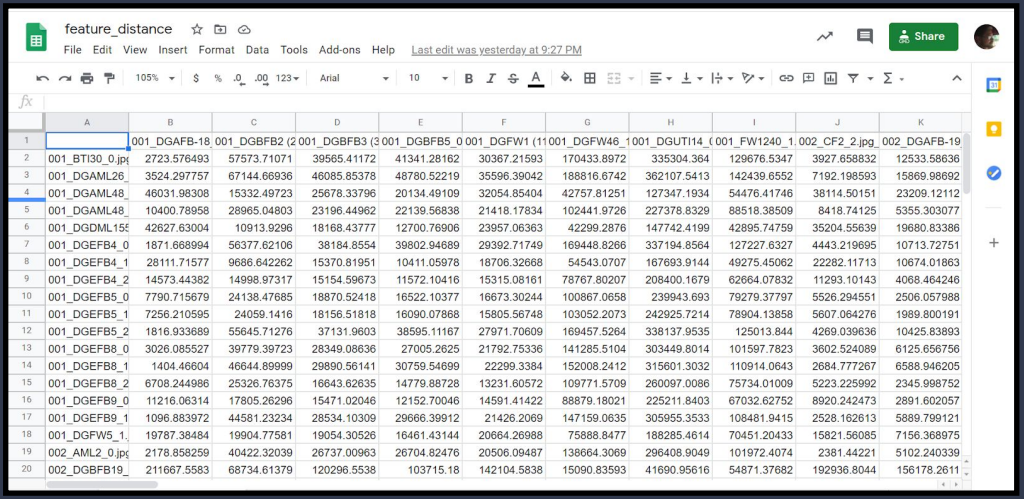
	A	B	C	D	E	F	G	H	I	J	K
1	001_DGAFB-18	2.845973	0.2568608	8.114507	16.412699	5.783505	1.4573632	3.852632	4.914076	13.013134	3.758657
2	001_DGBFB2 (2	0.25532255	9.907504	20.956127	25.955307	5.2036304	4.423435	7.532301	0.7025992	22.710056	4.7712507
3	001_DGBFB3 (3	1.3043972	9.740449	15.277913	26.133268	7.407022	8.539255	18.122877	0.039738335	21.328897	6.406266
4	001_DGBFB5_0	1.0366489	3.5902724	10.634958	8.894391	3.6741288	0.48015493	6.4394417	2.7806985	16.939981	6.556768
5	001_DGFW1 (11	0.3984795	11.120052	18.492739	26.568768	6.468583	5.183705	9.066516	2.0988898	24.643606	6.213155
6	001_DGFW46_1	4.7514825	13.101976	54.476463	24.62578	11.322186	5.232115	22.714376	6.9922705	33.984436	10.979604
7	001_DGUTI14_(	5.498443	170.8942	4.785547	104.61909	82.63592	0	14.59613	0	76.435425	2.121051
8	001_FW1240_1.	0.46812797	3.225378	34.900475	50.752007	0.9348731	7.330266	30.866566	5.7188187	50.587944	0.00723124
9	002_CF2_2.jpg_	1.0116638	0	105.36509	46.2274	26.181429	3.6792822	43.373848	39.041916	20.611616	33.340828
10	002_DGAFB-19_	2.1828907	0.708984	8.371278	45.48045	4.5771523	1.3292431	8.880152	0.5029392	18.01209	0.98844963
11	002_DGBFB2 (3	0.091975726	8.022797	15.2400055	73.83908	1.7252184	2.2839737	44.067562	0.38603806	52.83357	0.7505045
12	002_DGFB2 (17	1.0559125	69.93492	9.469738	61.98917	54.771168	0	11.998061	1.3246315	45.932224	0.61207294
13	002_DGFB2 (18	0.51676095	12.165291	21.339216	36.346424	6.8950505	3.215859	23.4095	0.778598	36.22133	1.4445678
14	002_DGFB2_1.j	4.8736305	0.55415577	52.657288	27.08926	9.008631	1.408498	17.102362	20.380745	21.588598	13.685191
15	002_DGFW31_1	2.4269407	8.858149	63.44425	56.109695	7.8043966	0	17.517824	4.413475	69.5719	2.9036887
16	002_DGGML2_(	2.7371838	8.2298565	15.357899	28.21141	4.582877	4.3004494	11.313791	1.9129093	16.4798	3.6060076
17	002_DGHFW2_(	1.9339161	1.8336142	21.66594	27.333942	4.0848722	2.0409164	6.460322	6.6417823	19.153343	11.264578
18	004_DFB49_2.j	1.8631505	0.5402077	20.105305	39.847256	10.149728	0.18579957	28.883612	9.592332	50.35773	5.4405074
19	004_DGAFB-21_	2.1150997	2.798519	3.6201239	13.187537	4.680111	1.1271769	3.262154	0.92975706	4.942541	5.0955315
20	005_DGFB9_2.j	5.371175	8.938274	34.27765	123.03831	28.964449	9.018929	61.733406	0.22474393	73.81221	5.299463

- On the left are the first 10 Feature vectors of the first 20 sample images we took.
- These are the feature vectors obtained for images of size 299*299 we have not displayed the feature vectors for the other set of images as it will be redundant, but have been included in the report.
- From quick scan through the vectors it was observed that Inceptionv3 had values in the columns with lower values, this makes me believe that that it was not successful in determining the features properly, as compared to Resnet50.

Distance or mean square error

```
disfeatures=[]
for i in range(0,100):
    dis=[]
    for j in range(0,73):
        add=0
        for k in range(0,2047):
            sub = (l2[i][0][k]-l[j][0][k])**2
            add = add + sub
        add=add/2048
        dis.append(add)
    disfeatures.append(dis)
```

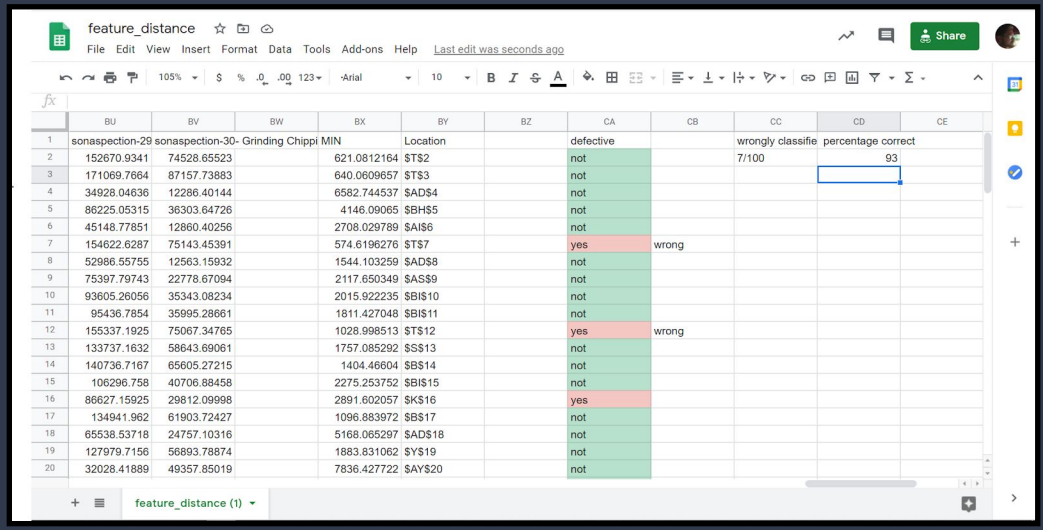
- On the left we have a code snippet of the code used for finding the distance or the mean square error.
- So each test image is taken which has 2048 features, similarly the sample labels have 2048 features, the respective features difference is found and then squared and it is added into a sum. This sum keep on increasing until the last features are subtracted squared and added.
- The final sum is divided by 2048 as to get an average value and this is repeated for the next sample image. And these process are repeated for all test images.
- Then the least of the distances is taken and it is classified as that image, and then we compared the images to see if they were right and wrong


feature_distance

File Edit View Insert Format Data Tools Add-ons Help Last edit was yesterday at 9:27 PM

Share

	A	B	C	D	E	F	G	H	I	J	K
1		001_DGAFB-18	001_DGBFB2 (2	001_DGBFB3 (3	001_DGBFB5_0	001_DGFW1 (11	001_DGFW46_1	001_DGUTI14_(	001_FW1240_1	002_CF2_2.jpg_	002_DGAFB-19
2	001_BTI30_0.jpg	2723.576493	57573.71071	39565.41172	41341.28162	30367.21593	170433.8972	335304.364	129676.5347	3927.658832	12533.58636
3	001_DGAML26_	3524.297757	67144.66936	46085.85378	48780.52219	35596.39042	188816.6742	362107.5413	142439.6552	7192.198593	15869.98692
4	001_DGAML48_	46031.98308	15332.49723	25678.33796	20134.49109	32054.85404	42757.81251	127347.1934	54476.41746	38114.50151	23209.12112
5	001_DGAML48_	10400.78958	28965.04803	23196.44962	22139.56838	21418.17834	102441.9726	227378.8329	88518.38509	8418.74125	5355.303077
6	001_DGDML155	42627.63004	10913.9296	18168.43777	12700.76906	23957.06363	42299.2876	147742.4199	42895.74759	35204.55639	19680.83386
7	001_DGEFB4_0	1871.668994	56377.62106	38184.8554	39802.94689	29392.71749	169448.8266	337194.8564	127227.6327	4443.219695	10713.72751
8	001_DGEFB4_1	28111.71577	9686.642262	15370.81951	10411.05978	18706.32668	54543.0707	167693.9144	49275.45062	22282.11713	10674.01863
9	001_DGEFB4_2	14573.44382	14998.97317	15154.59673	11572.10416	15315.08161	78767.80207	208400.1679	62664.07832	11293.10143	4068.464246
10	001_DGEFB5_0	7790.715679	24138.47685	18870.52418	16522.10377	16673.30244	100867.0658	239943.693	79279.37797	5526.294551	2506.057988
11	001_DGEFB5_1	7256.210595	24059.1416	18156.51818	16090.07868	15805.56748	103052.2073	242925.7214	78904.13858	5607.064276	1989.800191
12	001_DGEFB5_2	1816.933689	55645.71276	37131.9603	38595.11167	27971.70609	169457.5264	338137.9535	125013.844	4269.039636	10425.83893
13	001_DGEFB8_0	3026.085527	39779.39723	28349.08636	27005.2625	21792.75336	141285.5104	303449.8014	101597.7823	3602.524089	6125.656756
14	001_DGEFB8_1	1404.46604	46644.89999	29890.56141	30759.54699	22299.3384	152008.2412	315601.3032	110914.0643	2684.777267	6588.946205
15	001_DGEFB8_2	6708.244986	25326.76375	16643.62635	14779.88728	13231.60572	109771.5709	260097.0086	75734.01009	5223.225992	2345.998752
16	001_DGEFB9_0	11216.06314	17805.26296	15471.02046	12152.70046	14591.41422	88879.18021	225211.8403	67032.62752	8920.242473	2891.602057
17	001_DGEFB9_1	1096.883972	44581.23234	28534.10309	29666.39912	21426.2069	147159.0635	305955.3533	108481.9415	2528.162613	5889.799121
18	001_DGFW5_1_	19787.38484	19904.77581	19054.30526	16461.43144	20664.26988	75888.8477	188285.4614	70451.20433	15821.56085	7156.368975
19	002_AML2_0.jpg	2178.858259	40422.32039	26737.00963	26704.82476	20506.09487	138664.3069	296408.9049	101972.4074	2381.44221	5102.240339
20	002_DGBFB19_	211667.5583	68734.61379	120296.5538	103715.18	142104.5838	15090.83593	41690.95616	54871.37682	192936.8044	156178.2611


feature_distance

File Edit View Insert Format Data Tools Add-ons Help Last edit was seconds ago

Share

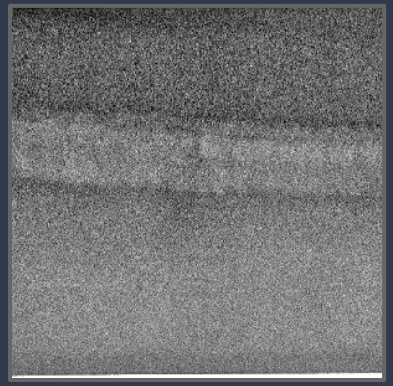
	BU	BV	BW	BX	BY	BZ	CA	CB	CC	CD	CE
1	sonaspection-29	sonaspection-30-	Grinding Chippi	MIN	Location		defective		wrongly classifie	percentage correct	
2	152670.9341	74528.65523		621.0812164	T2		not		7/100	93	
3	171069.7664	87157.73883		640.0609657	T3		not				
4	34928.04636	12286.40144		6582.744537	AD4		not				
5	86225.05315	36303.64726		4146.09065	BH5		not				
6	45148.77851	12860.40256		2708.029789	AI6		not				
7	154622.6287	75143.45391		574.6196276	T7		yes	wrong			
8	52986.55755	12563.15932		1544.103259	AD8		not				
9	75397.79743	22778.67094		2117.650349	AS9		not				
10	93605.26056	35343.08234		2015.922235	BI10		not				
11	95436.7854	35995.28661		1811.427048	BI11		not				
12	155337.1925	75067.34765		1028.998513	T12		yes	wrong			
13	133737.1632	58643.69061		1757.085292	S13		not				
14	140736.7167	65605.27215		1404.46604	B14		not				
15	106296.758	40706.88458		2275.253752	BI15		not				
16	86627.15925	29812.09998		2891.602057	K16		yes				
17	134941.962	61903.72427		1096.883972	B17		not				
18	65538.53718	24757.10316		5168.065297	AD18		not				
19	127979.7156	56893.78874		1883.831062	Y19		not				
20	32028.41889	49357.85019		7836.427722	AY20		not				

feature_distance (1)

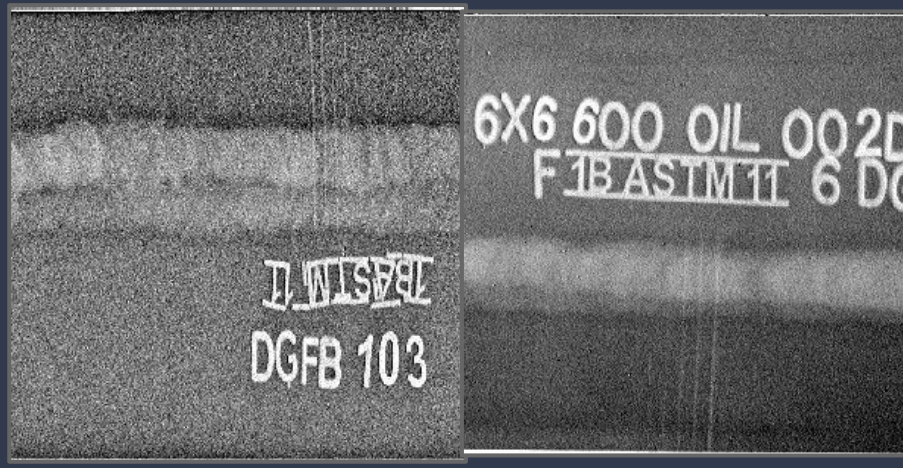
Final Results

	299*299_resnet50	299*500_resnet50	299*299_inception_v3	299*500_inception_v3
#Defective samples	21	24	21	24
#Non defective	79	76	79	76
#Wrongly classified	7	6	8	6
Accuracy %	93	94	92	94

Correctly Classified



Sample Defect



Wrongly Classified

Confusion Matrix

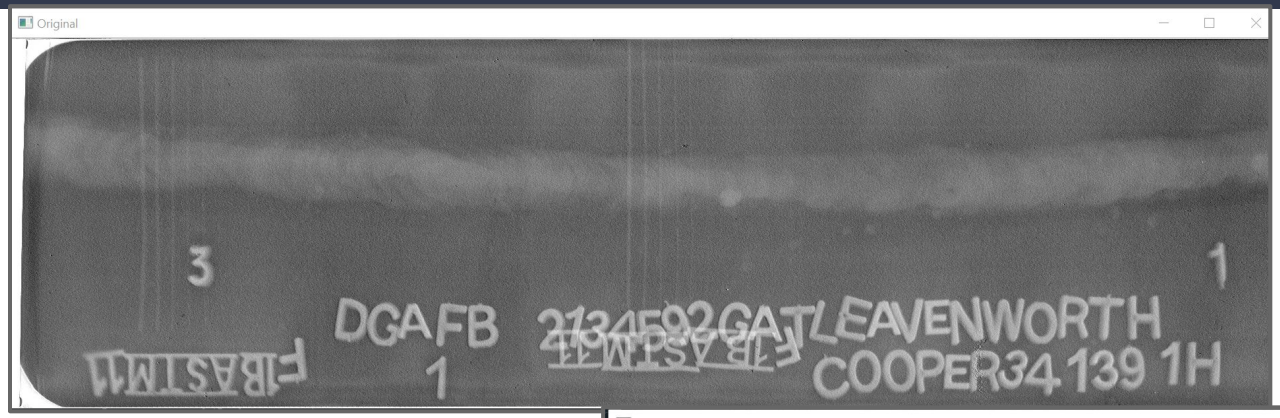
299*299_Resnet50	Predicted Non-Defective	Predicted Defective
Actual Non-Defective	79	0
Actual Defective	7	14

299*500_Resnet50	Predicted Non-Defective	Predicted Defective
Actual Non-Defective	76	0
Actual Defective	6	18

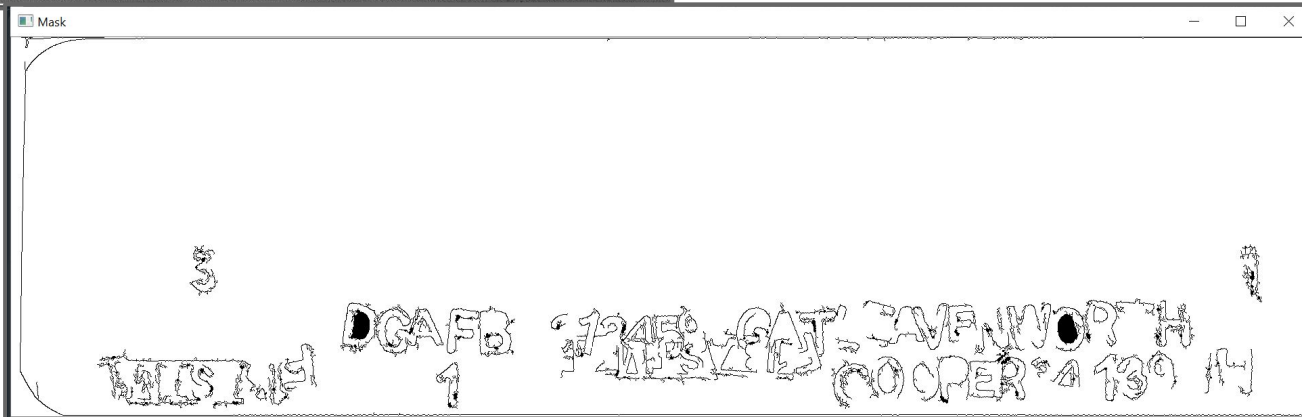
299*299_Inception_v3	Predicted Non-Defective	Predicted Defective
Actual Non-Defective	79	0
Actual Defective	8	13

299*500_Inception_v3	Predicted Non-Defective	Predicted Defective
Actual Non-Defective	76	0
Actual Defective	6	18

Future work



We look to remove the letters present on the images so that they do not wrongly classify the image



Reference paper links

1. *Tensorflow hub from google*

2. *Keras from google*

3. *Imagenet models- ResNet50,Inception_v3*

4. Moghaddam, Alireza Azari (2015). *[IEEE 2015 2nd International Conference on Knowledge-Based Engineering and Innovation (KBEI) - Tehran, Iran (2015.11.5-2015.11.6)] 2015 2nd International Conference on Knowledge-Based Engineering and Innovation (KBEI) - Image processing technique for classification of linear welding defects..*

Thank You!!