Hochschule
Bonn-Rhein-Sieg
University of Applied Sciences

Bonn-Aachen
International Center for
Information Technology

R&D Project

Benchmarking Uncertainty Estimation of Deep Learning Models Using Synthetic Dataset

Sathwik Panchangam

Submitted to Hochschule Bonn-Rhein-Sieg,
Department of Computer Science
in partial fullfilment of the requirements for the degree
of Master of Science in Autonomous Systems

Supervised by

Prof. Dr. Nico Hochgeschwender
MSc. Deebul Nair

October 2022

I, the undersigned below, declare that this work has not previously been submitted to this or any other university and that it is, unless otherwise stated, entirely my own work.

Date	Sathwik Panchangam

Abstract

Deep learning models are well known for their stochastic nature and they provide point estimates with no measure of uncertainty which leads to overconfident predictions and misclassifications.

Acknowledgements

Thanks to ....

Contents

List of Figures

List of Tables

1

Introduction

Estimating uncertainty in the predictions of deep learning models is very important to avoid overconfident predictions and misclassifications.

1.1 Problem Statement

Write about proper problem statement. dont write about the approach

Helpful for the testers. I want my model to fail in some cases.

Defining the ground truth for the uncertainty is difficult and challenging because there is no metric for it

also speak if there are any other challenges just overview.

dont include about fuzzy logic here.Speak about it later

1.2 RnD Stucture

Chapter2 fhsaoihgf. Chapter3 adoskf. Chapter4 sfhashf. Chapter5 ajfshs. etc go on.

1.3 Contributions

2

State of the Art

2.1 Blender for synthetic data generation

Write about what kind of synthetic data can be generated using blender

2.2 Uncertainty

In deep learning uncertainty can be defined as a partial or complete lack of knowledge of the outcome of the predictions made by the deep learning models [**?**]. Uncertainty in deep learning models can be classified into two types.

- Aleatoric Uncertainty: This type of uncertainty is caused due to the presence of uncertainty in the training data.
- Epistemic Uncertainty: This type of uncertainty is caused due to the lack of knowledge of the neural network.

One of the most common ways to estimate the predictive uncertainty in deep learning models is to model the epistemic and aleatoric uncertainties separately.

2.3 Uncertainty Estimation

....

2.3.1 Use of synthetic data in deep learning and uncertainty estimation

......

2.3.2 Comparison of state-of-the-art uncertainty estimation techniques

.....

Use as many sections as you need in your related work to group content into logical groups

Don't forget to correctly cite your sources [1].

2.4 Limitations of previous work

3 Methodology

3.1 Image Rendering in blender

Write about blender api python and how you can generate images.

3.2 Defining the expected value for uncertainty

Write about fuzzy logic

3.3 Synthetic data generation

Write about the ways you want to generate the data.
or modifiable parameters in blender (think of final goal)

3.4 Deep Learning

3.4.1 ResNet

Write about skip connections and also add an image.

3.4.2 CrossEntropyLoss

end with why cross entropy loss is not good for uncertainty because of softmax

3.4.3 Uncertainty Estimation

to avoid softmax errors uncertainty is helpful.
end with Which uncertainty estimation techniques?

3.4.3.1 Evidential Loss

How you are planning to test/compare/evaluate your research. Criteria used.

3.5 Setup

- Just write about the overview of the process dont write the actual method. write it in the next session
- Write about generating dataset in blender
- Write about fuzzy logic
- Write about training and estimating uncertainty and final comparison use the block diagrams or the diagrams available on github and proposal you may need to modify them.

3.6 Experimental Design

4
Solution

4.1 Proposed approach

... Need to write this section

4.2 Implementation details

For generating the synthetic data, 3D cad models of all the 15 objects of RoboCup@work components present in 5.1 were designed and appropriate materials and textures were applied such that the models will look similar to the real world objects. The figure 4.1 shows the 3D cad models of the objects. These models were then used to generate different synthetic datasets using the blender's API (Bpy).

Figure 4.1: 3D cad models of the objects

4.2.1 Modifiable Parameters

With the help of Blender's api we can control different parameters of the software to introduce different variations to the synthetic dataset. Some of the modifiable parameters used in this project are discussed in this section.

4.2.1.1 Rendering

Before going into the modifyable parameters it is important to understand what is rendering and how is it done in blender. Blender has two inbuilt rendering engines, Evee and Cycles both of them aim to be similar in view but works completely different.

The main priority of Evee rendering engine is speed and hence it is appropriate for close to real-time performance. On the other hand Cycles is a ray-traced render engine which is much slower but has an advantage of producing light conditions similar to how the light bounces in the real-world [2].

4.2.1.2 Random lighting

For the lighting setup in blender, among the available light types of Point, Area, Spot and Sun we have considered the light type of Sun. The main reason for choosing the Sun type is due one of the constraint of the proposed approach which will be discussed in the section 4.2.2.

In blender, the Sun object is placed at an arbitrarily choosen position and orientation such that all the objects present in the scene are properly visible. The default range of the intensity of light (Sun object) is considered i.e (0-10). Here the value 0 produces dark lighting conditions and the value 10 produces bright lighting conditions as shown in fig 4.2. With in this range a random value is choosen for each image while rendering.

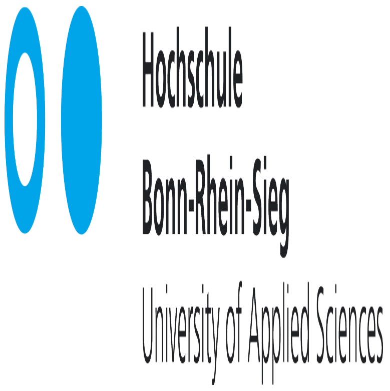


Figure 4.2: Lighting conditions

4.2.1.3 Random rotation

For the random rotation of the object Blender requires the input value to be in radians and it takes euler angles as an input, so a random value is choosen between 0 and 1 and then it is multiplied by 2 * Π to get a value in radians and provided as euler angles. In this way a random value is choosen for orientations along x,y and z. The origin of each object is set to its center of mass. All the orientations takes place with respect to the axis of the origin of the object and not with respect to the global origin.

This kind of rotation is done because of two reasons. First reason is, the scene setup in bender is created as a background plane with objects on the top. Here if the objects are rotated with respect to the global origin then they take large rotation angles which results in a senario where the objects go behind the background plane and will not be visible in the image during the rendering process.

The second reason is, due this large rotation angles even if we restrict the rotation of the object to some default ranges and track the camera to the view of the object, the area of the background needs to be scaled up so that the object will be on top of the plane. This scaling will result in high computation and rendering time as all the mesh objects visible in the scene will be taken into consideration during rendering process.

Figure 4.3: Random Rotation of the objects

4.2.1.4 Random background

For the random background we have used two methods. In the first method the we change the base color of the PrincipledBSDF node in the background plane's object. This base color represents RGB color space and expects a tuple of 4 values containing r,g,b and alpha values. For the whole experiments the alpha value is kept as a constant value of one. A random value will be choosen for (r,g,b) for each image while rendering with the help of blender's api. Some of the examples for the random colors can be seen in the fig 4.4a.

In the second method, images of the desired background textures were provided in a folder and the a random image is choosen from the available images. This image is passed as a material to the background plane with the help of image texture node. Some of the examples of the images with random materials are shown in the fig 4.4b

Hochschule
Bonn-Rhein-Sieg
University of Applied Sciences
Hochschule
Bonn-Rhein-Sieg
University of Applied Sciences

(a) Random color backgrounds (b) Random material backgrounds

Figure 4.4: Random Backgrounds

4.2.1.5 Resolution

For changing the resolution of the image blender expects a tuple of three values containing the resolution of value of x, resolution value of y and percentage scale of the resolution is set to 100 % by default. Only the values of x and y are provided as integer values to generate the image with desired resolution.

4.2.1.6 Random focal length

...

4.2.1.7 Blur

......

4.2.1.8 Occulusion

......

4.2.1.9 Random materials for the objects

......

4.2.2 Computing expected uncertainty value

Mention about why did you choose sun type ? reason fuzzy logic to create expected value. and material threshold dictionary

also write about how you are saving the uncertainty label. json file and what other parameters you are adding to that file.

4.3 Challenges

5

Evaluation and Results

5.1 Experiment Description

In this work, we have performed two different experiments to test the performance of deep learning models. The first experiment is using the proposed approach of defining the expected uncertainty value based on Fuzzy logic. The second experiment is to generate synthetic data by varying the lighting conditions between the regions of bright, normal and dark. These two methods are discussed in detail in the following experiments section.

5.2 Experimental Setup

The synthetic datasets used in the experiments are generated using the software blender version 3.2.0 and its rendering capabilities. All the images are rendered with a resolution of 96 x 96 and 4096 samples for each point in cycles render engine.

5.2.1 Dataset

In this work, we have considered the RoboCup@work components dataset as the primary dataset which consists of 15 categories of general industrial objects as shown in the figure 5.1. The dataset contains a total of 21706 images of real world objects.

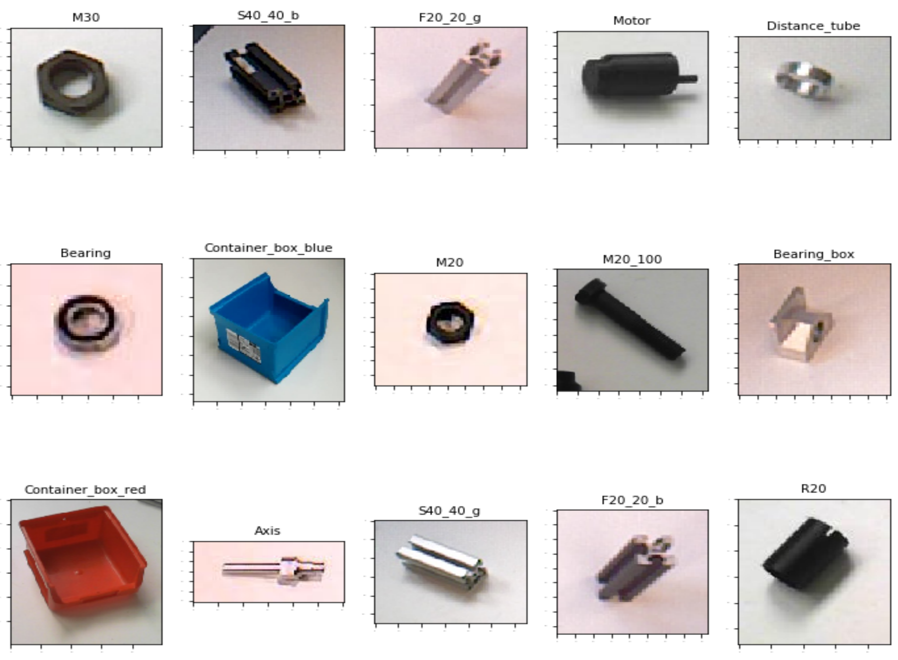


Figure 5.1: RoboCup@work objects

5.2.2 Synthetic Dataset Generation

For generating the synthetic data for testing the performance of the deep learning models we use the modifiable parameters discussed in the previous section and generate different datasets based on the type of experiments that we perform.

5.2.2.1 Dataset with expected uncertainty value

The main aim of creating this dataset is to define a expected value for the uncertainty in the image based on different lighting conditions. To achieve this, we use the methodology discussed in the section 4.1 which is based on fuzzy logic. Initially we have imported 3d models of all the required objects along with their materials and textures into the scene created in blender. This scene consists of a background plane, a camera and a light source of type sun.

For the background plane a default color is choosen from RGB color space of the Principled BSDF node of the material belonging to the background plane object. Since this experiment focuses on defining the expected value for the uncertainty in different lighting conditions we do not change the color or material of the background plane. Another reason is if we change the color or material of the background plane we may need to define the fuzzy rules for this change aswell to calculate the expected value of the uncertainty. This is because we know that the performance of the DNN model will be affected with change in the background.

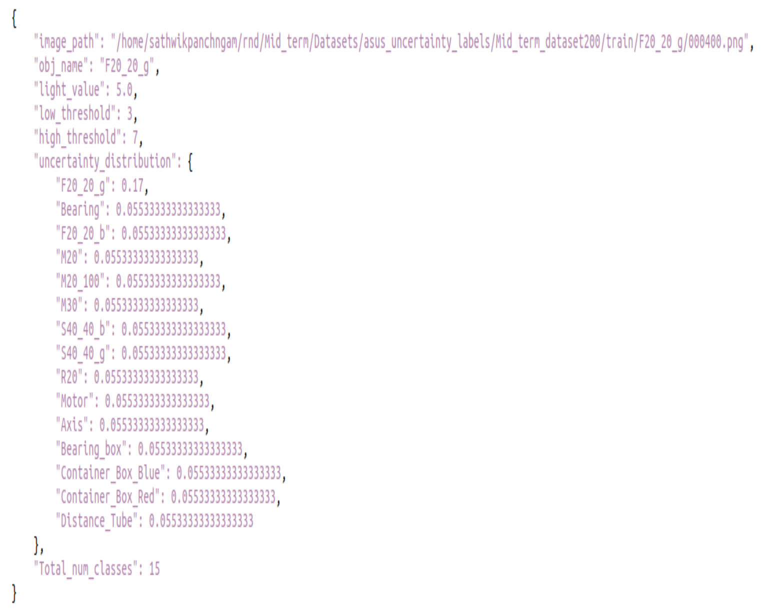
The camera object was placed in the scene such that all the objects will be visible in the render frame. Then random rotation and random focal length functions discussed in the sections 4.2.1.3 , 4.2.1.6 are used to generate data with different orientations and different perspectives of the objects.

The light source of type sun is used to provide different lighting conditions during the rendering process and the intensity of this light source is set between the range of 0 and 10 which is the default range of blender. Then random lighting functionality from 4.2.1.2 was used to provide different variations.

After setting up the camera parameters and light parameters we have computed the uncertainty for each image based on the methodology discussed in the section 4.2.2. Then finally we have generated the dataset with 200 images for each object along with their expected uncertainty value as an uncertainty label. Thus by this approach we have created a dataset in the below form.

$$data = (image, class\_label, uncertaitny\_label)$$

An example image of the uncertainty label can be seen in 5.2 and the rendered images along with their class labels can be seen in the figure 5.3 .



```
{
    "image_path": "/home/sathwikpanchngam/rnd/Mid_term/Datasets/asus_uncertainty_labels/Mid_term_dataset200/train/F20_20_g/000400.png",
    "obj_name": "F20_20_g",
    "light_value": 5.0,
    "low_threshold": 3,
    "high_threshold": 7,
    "uncertainty_distribution": {
        "F20_20_g": 0.17,
        "Bearing": 0.05533333333333333,
        "F20_20_b": 0.05533333333333333,
        "M20": 0.05533333333333333,
        "M20_100": 0.05533333333333333,
        "M30": 0.05533333333333333,
        "S40_40_b": 0.05533333333333333,
        "S40_40_g": 0.05533333333333333,
        "R20": 0.05533333333333333,
        "Motor": 0.05533333333333333,
        "Axis": 0.05533333333333333,
        "Bearing_box": 0.05533333333333333,
        "Container_Box_Blue": 0.05533333333333333,
        "Container_Box_Red": 0.05533333333333333,
        "Distance_Tube": 0.05533333333333333
    },
    "Total_num_classes": 15
}
```

Figure 5.2: Example image of uncertainty label in json format

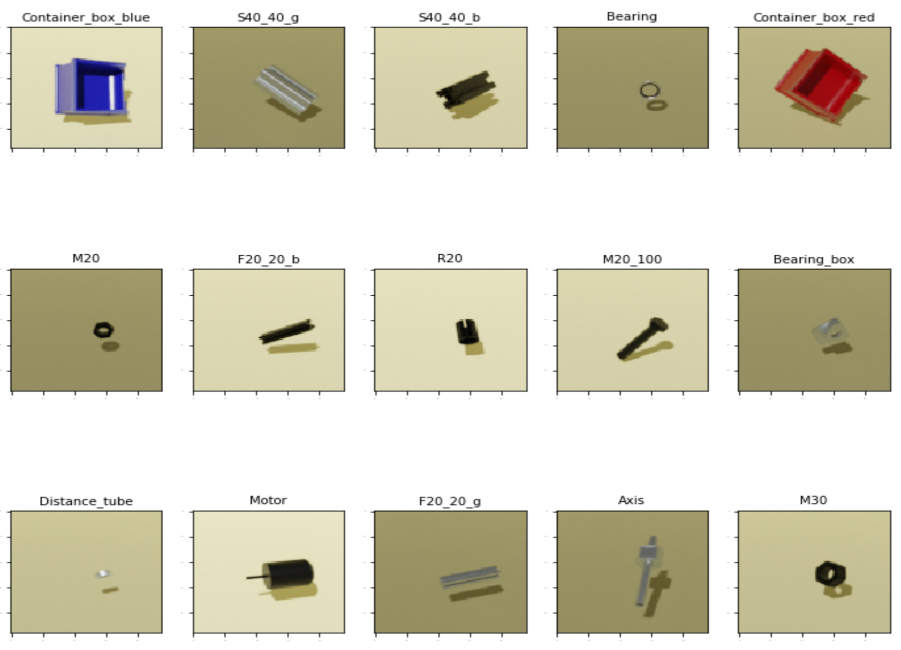


Figure 5.3: Image with uncertainty label

5.2.2.2 Dataset for testing on light conditions

The primary goal of creating this type of dataset is to test the performance of the deep learning model in different lighting conditions such as bright, normal and dark. Here the dataset will be similar to a classification dataset with images and class labels but the data will be generated in three scenarios (bright, normal and dark).

To achieve this, the random lighting functionality discussed in the section 4.2.1.2 , is used and a threshold for the lighting is defined. If the value of the light in blender is less than or equal to 2 then the lighting is considered as dark. If the light value is greater than or equal to 10 then the lighting is considered as bright. The region between the values of 2 and 10 is considered as normal lighting region.

The threshold here is considered by observing the scene in different light values and the value at which the images render with dark or black color is choosen a min threshold and the bright or white color is choosen as max threshold.

In addition to the lighting, random rotation functionality 4.2.1.3 is added to the objects and from random background functionality 4.2.1.4 random colors are added to the background plane.

Some of the examples of the bright, normal and dark images are shown in the fig 5.4.

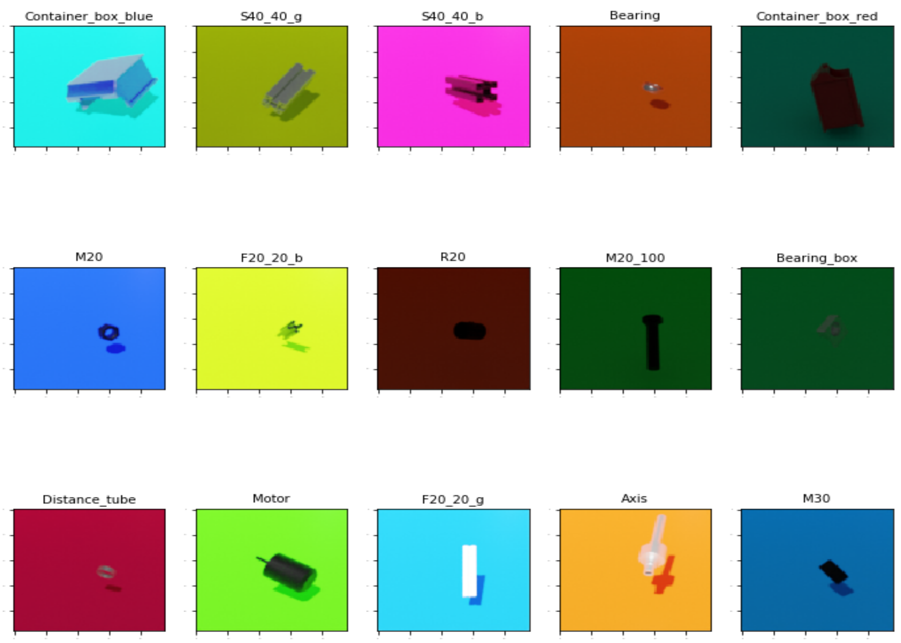


Figure 5.4: Different Lighting

5.2.3 Uncertainty estimation

The objective of all the experiments conducted in this project is to test the performance of the deep learning model in a classification task using different uncertainty estimation techniques.

For this purpose a Resnet18 model with default ImageNet pretrained weights is considered as the deep learning model for all the experiments. Only the number of output neurons in the final layer is changed to the number of classes present in the dataset. i.e for RoboCup@work components - 15 classes.

In this experiment the ResNet 18 model is trained with two loss functions crossentropy loss and evidential loss. For training the model the hyperparameters provided in table 5.1 were used.

After training both the models with best accuracy are saved as .pth files and they are used again to validate the model on the test dataset for one run.

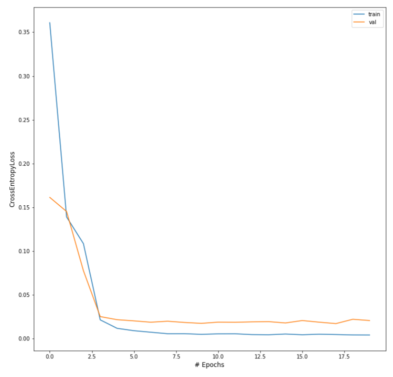
Hyperparameters	
Model	ResNet18
Number of epochs	20
Batch size	64
Learning Rate	1e-3
Weight Decay	1e-5
Optimizer	Adam

Table 5.1: Hyperparameters

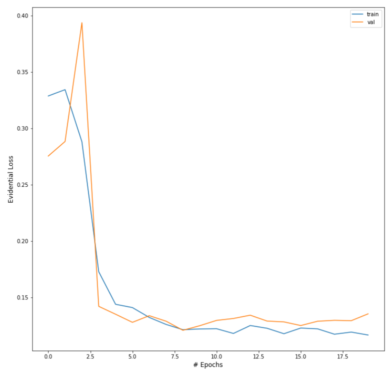
The model is trained for 20 epochs using StepLR scheduler with a learning rate of 1e-3 and a weight deacy of 1e-5.

5.2.3.1 Testing with real world dataset

The loss of the model during training and validation with both CrossEntropy loss and EvidentialLoss is depicted in the figure 5.5a and 5.5b .



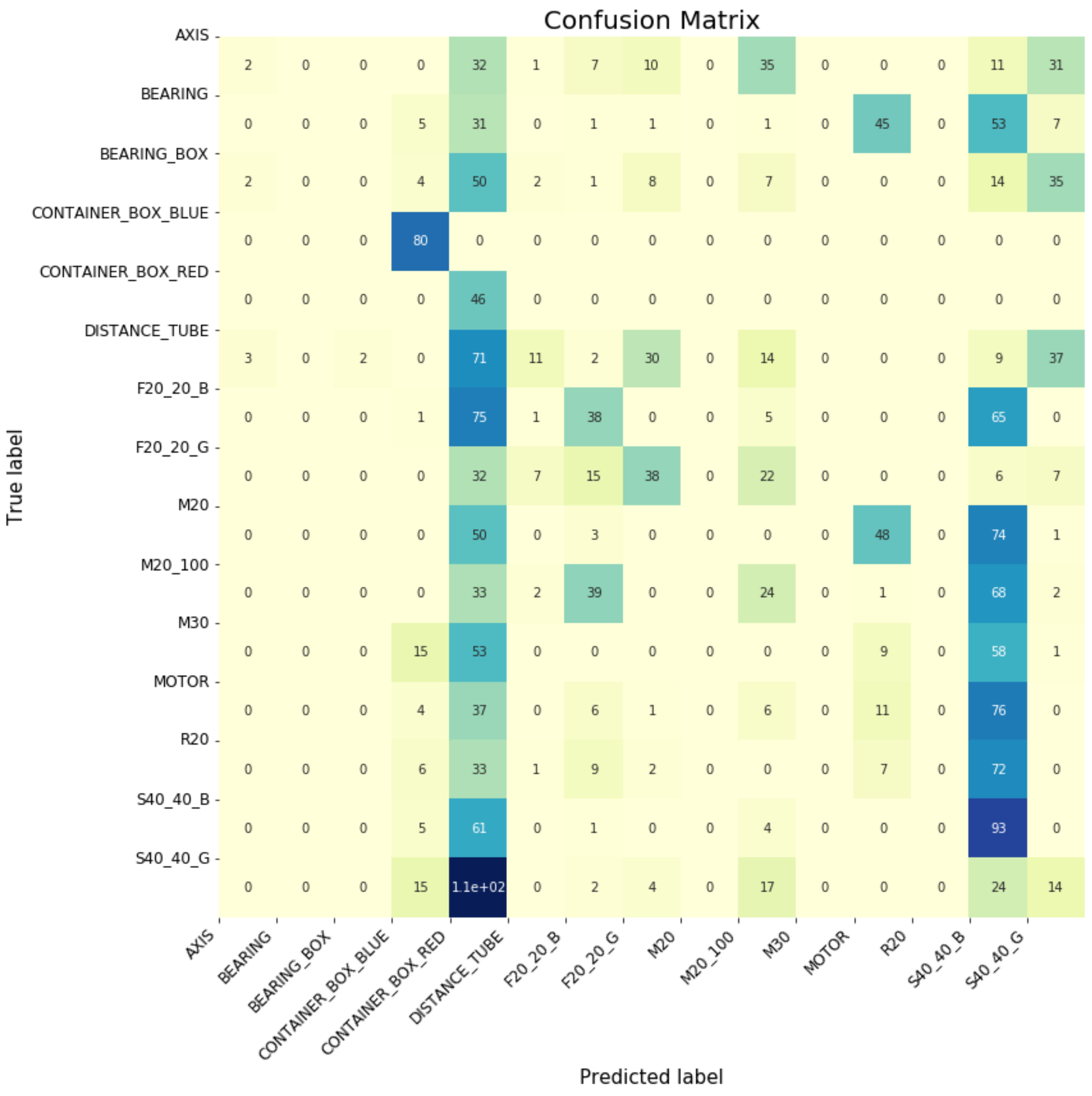
(a) Loss comparison for cross entropy



(b) Loss comparison for evidentaial

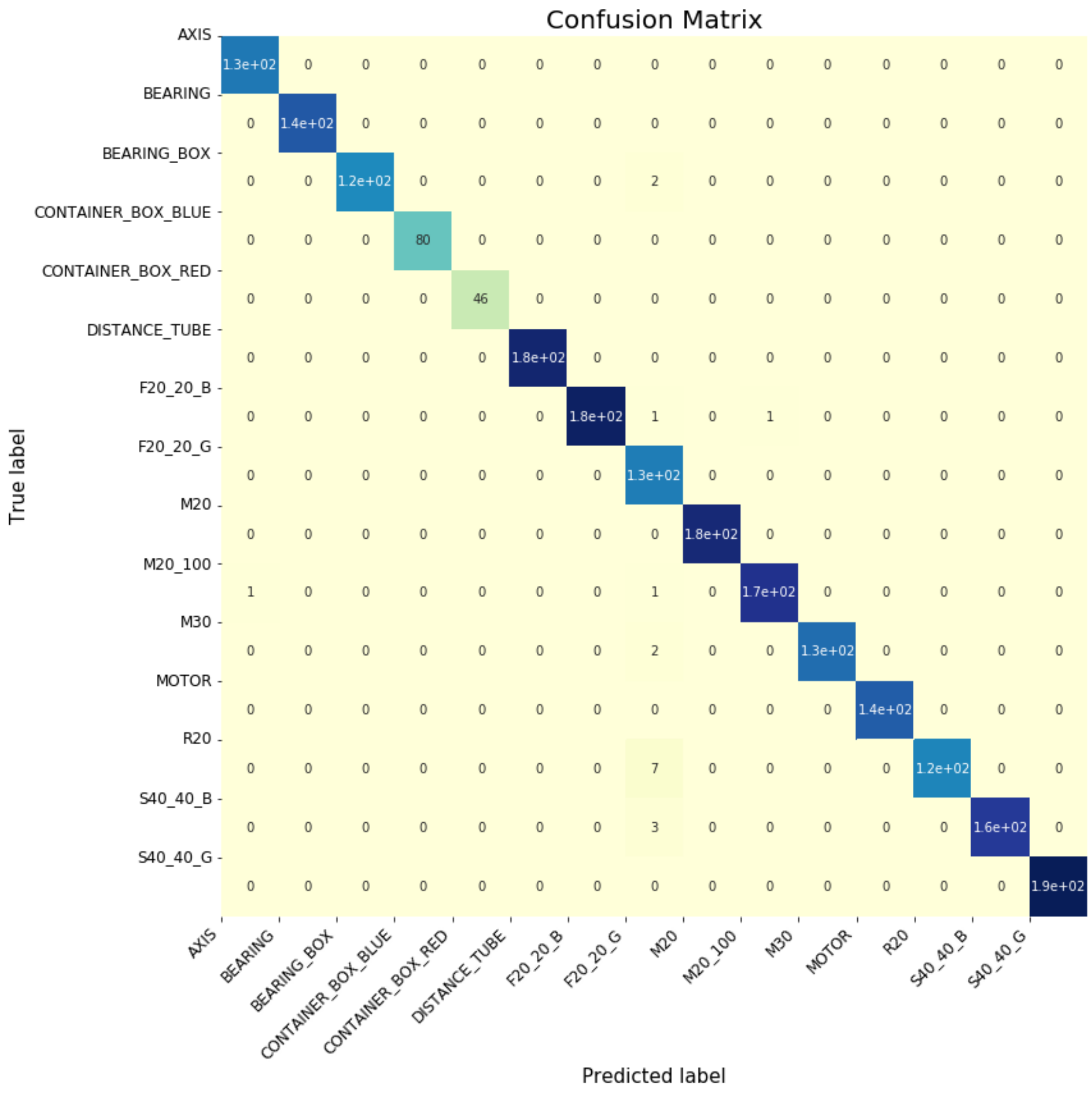
Figure 5.5: Training and validation losses for original dataset

The confusion matrix for the real world test data using crossentropy model is shown in the figure 5.6 and with evidential loss is shown in the figure 5.7



True label \ Predicted label	AXIS	BEARING	BEARING_BOX	CONTAINER_BOX_BLUE	CONTAINER_BOX_RED	DISTANCE_TUBE	F20_20_B	F20_20_G	M20	M20_100	M30	MOTOR	R20	S40_40_B	S40_40_G
AXIS	2	0	0	0	32	1	7	10	0	35	0	0	0	11	31
BEARING	0	0	0	5	31	0	1	1	0	1	0	45	0	53	7
BEARING_BOX	2	0	0	4	50	2	1	8	0	7	0	0	0	14	35
CONTAINER_BOX_BLUE	0	0	0	80	0	0	0	0	0	0	0	0	0	0	0
CONTAINER_BOX_RED	0	0	0	0	46	0	0	0	0	0	0	0	0	0	0
DISTANCE_TUBE	3	0	2	0	71	11	2	30	0	14	0	0	0	9	37
F20_20_B	0	0	0	1	75	1	38	0	0	5	0	0	0	65	0
F20_20_G	0	0	0	0	32	7	15	38	0	22	0	0	0	6	7
M20	0	0	0	0	50	0	3	0	0	0	0	48	0	74	1
M20_100	0	0	0	0	33	2	39	0	0	24	0	1	0	68	2
M30	0	0	0	15	53	0	0	0	0	0	0	9	0	58	1
MOTOR	0	0	0	4	37	0	6	1	0	6	0	11	0	76	0
R20	0	0	0	6	33	1	9	2	0	0	0	7	0	72	0
S40_40_B	0	0	0	5	61	0	1	0	0	4	0	0	0	93	0
S40_40_G	0	0	0	15	1.1e+02	0	2	4	0	17	0	0	0	24	14

Figure 5.6: Confusion matrix of real world test data with cross entropy loss



	AXIS	BEARING	BEARING_BOX	CONTAINER_BOX_BLUE	CONTAINER_BOX_RED	DISTANCE_TUBE	F20_20_B	F20_20_G	M20	M20_100	M30	MOTOR	R20	S40_40_B	S40_40_G
AXIS	1.3e+02	0	0	0	0	0	0	0	0	0	0	0	0	0	0
BEARING	0	1.4e+02	0	0	0	0	0	0	0	0	0	0	0	0	0
BEARING_BOX	0	0	1.2e+02	0	0	0	0	2	0	0	0	0	0	0	0
CONTAINER_BOX_BLUE	0	0	0	80	0	0	0	0	0	0	0	0	0	0	0
CONTAINER_BOX_RED	0	0	0	0	46	0	0	0	0	0	0	0	0	0	0
DISTANCE_TUBE	0	0	0	0	0	1.8e+02	0	0	0	0	0	0	0	0	0
F20_20_B	0	0	0	0	0	0	1.8e+02	1	0	1	0	0	0	0	0
F20_20_G	0	0	0	0	0	0	0	1.3e+02	0	0	0	0	0	0	0
M20	0	0	0	0	0	0	0	0	1.8e+02	0	0	0	0	0	0
M20_100	1	0	0	0	0	0	0	1	0	1.7e+02	0	0	0	0	0
M30	0	0	0	0	0	0	0	2	0	0	1.3e+02	0	0	0	0
MOTOR	0	0	0	0	0	0	0	0	0	0	0	1.4e+02	0	0	0
R20	0	0	0	0	0	0	0	7	0	0	0	0	1.2e+02	0	0
S40_40_B	0	0	0	0	0	0	0	3	0	0	0	0	0	1.6e+02	0
S40_40_G	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1.9e+02

Figure 5.7: Confusion matrix of real world test data with evidential loss

The most confidant and least confidant predictions of the model with cross entropy loss and evidential loss are depicted in the figures 5.8 , 5.9 , 5.10 , 5.11 .

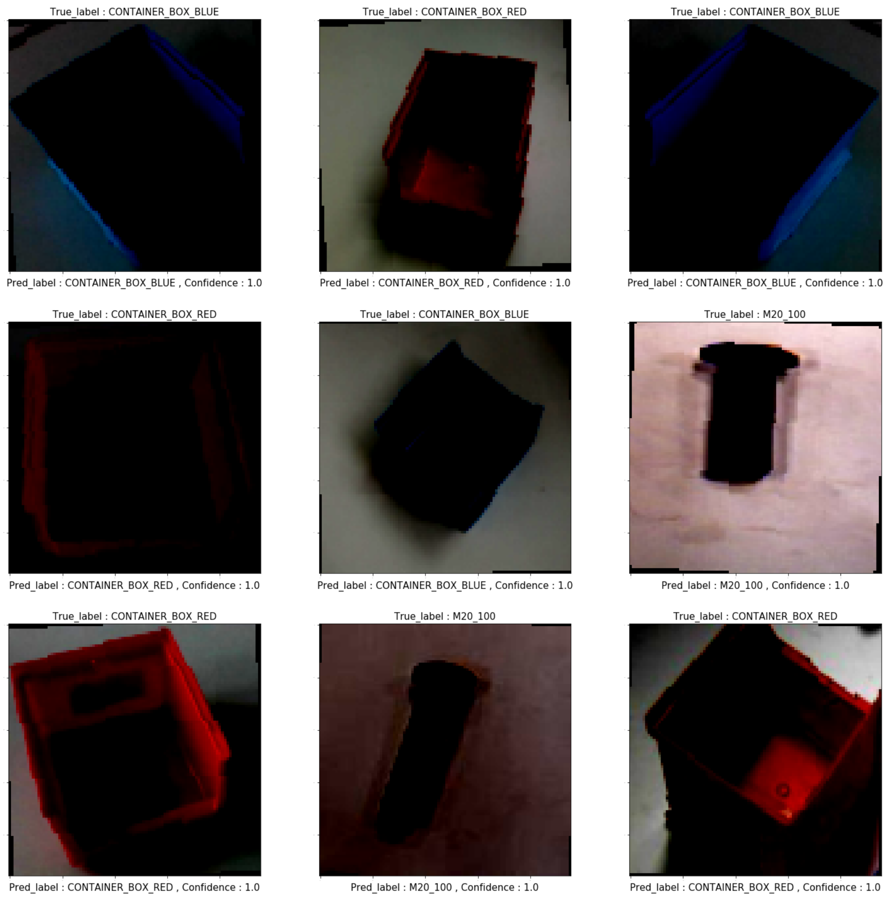

True_label : CONTAINER_BOX_BLUE
Pred_label : CONTAINER_BOX_BLUE , Confidence : 1.0
True_label : CONTAINER_BOX_RED
Pred_label : CONTAINER_BOX_RED , Confidence : 1.0
True_label : CONTAINER_BOX_BLUE
Pred_label : CONTAINER_BOX_BLUE , Confidence : 1.0
True_label : CONTAINER_BOX_RED
Pred_label : CONTAINER_BOX_RED , Confidence : 1.0
True_label : CONTAINER_BOX_BLUE
Pred_label : CONTAINER_BOX_BLUE , Confidence : 1.0
True_label : M20_100
Pred_label : M20_100 , Confidence : 1.0
True_label : CONTAINER_BOX_RED
Pred_label : CONTAINER_BOX_RED , Confidence : 1.0
True_label : M20_100
Pred_label : M20_100 , Confidence : 1.0
True_label : CONTAINER_BOX_RED
Pred_label : CONTAINER_BOX_RED , Confidence : 1.0

Figure 5.8: Best predictions with cross entropy loss

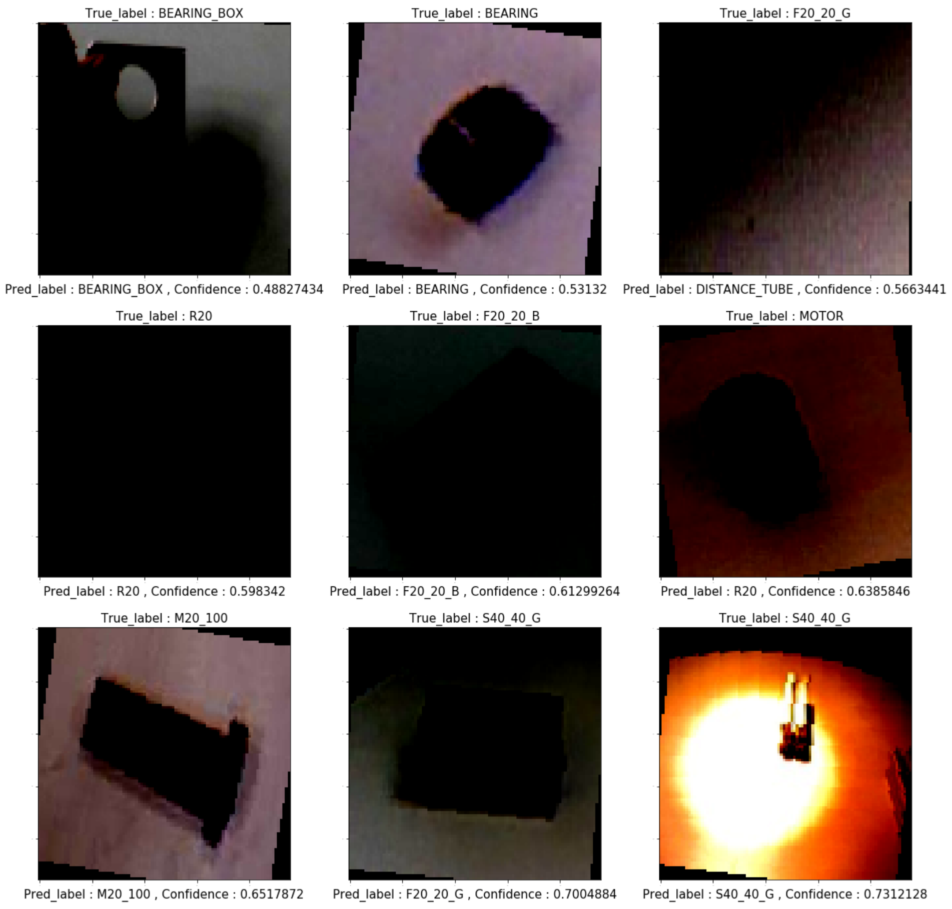

True_label : BEARING_BOX
Pred_label : BEARING_BOX , Confidence : 0.48827434
True_label : BEARING
Pred_label : BEARING , Confidence : 0.53132
True_label : F20_20_G
Pred_label : DISTANCE_TUBE , Confidence : 0.5663441
True_label : R20
Pred_label : R20 , Confidence : 0.598342
True_label : F20_20_B
Pred_label : F20_20_B , Confidence : 0.61299264
True_label : MOTOR
Pred_label : R20 , Confidence : 0.6385846
True_label : M20_100
Pred_label : M20_100 , Confidence : 0.6517872
True_label : S40_40_G
Pred_label : F20_20_G , Confidence : 0.7004884
True_label : S40_40_G
Pred_label : S40_40_G , Confidence : 0.7312128

Figure 5.9: Worst predictions with cross entropy loss

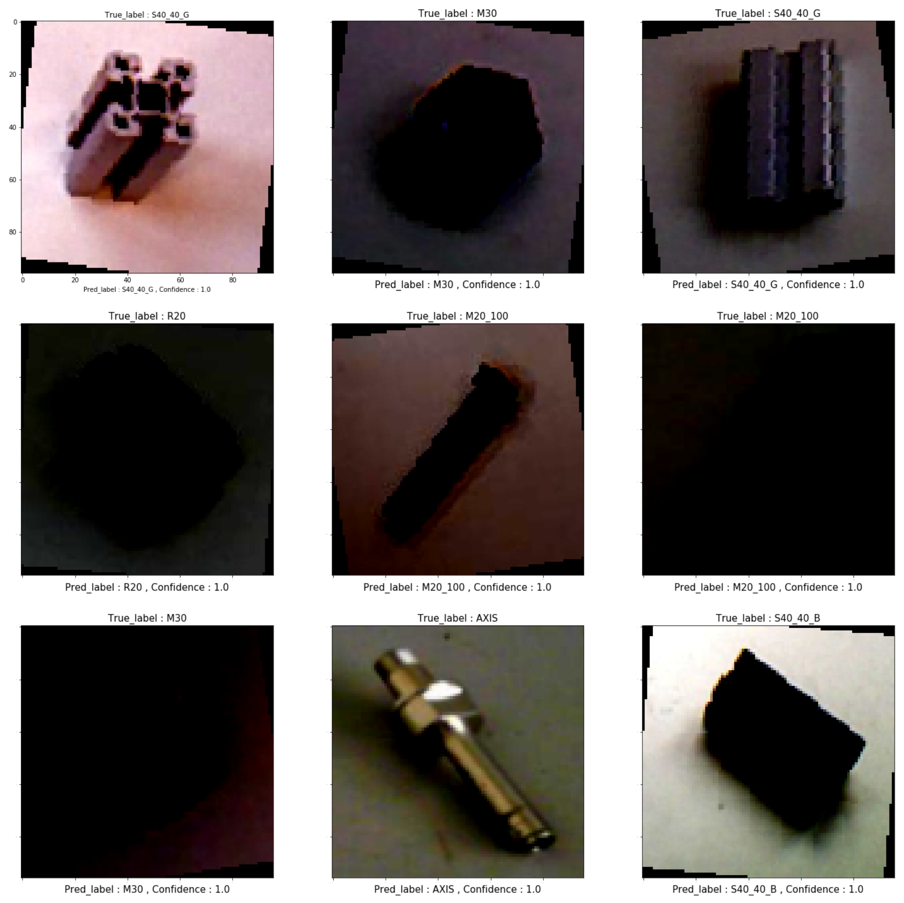

True_label : S40_40_G
0
20
40
60
80
0
20
40
60
80
Pred_label : S40_40_G , Confidence : 1.0
True_label : M30
Pred_label : M30 , Confidence : 1.0
True_label : S40_40_G
Pred_label : S40_40_G , Confidence : 1.0
True_label : R20
Pred_label : R20 , Confidence : 1.0
True_label : M20_100
Pred_label : M20_100 , Confidence : 1.0
True_label : M20_100
Pred_label : M20_100 , Confidence : 1.0
True_label : M30
Pred_label : M30 , Confidence : 1.0
True_label : AXIS
Pred_label : AXIS , Confidence : 1.0
True_label : S40_40_B
Pred_label : S40_40_B , Confidence : 1.0

Figure 5.10: Best predictions with evidential loss

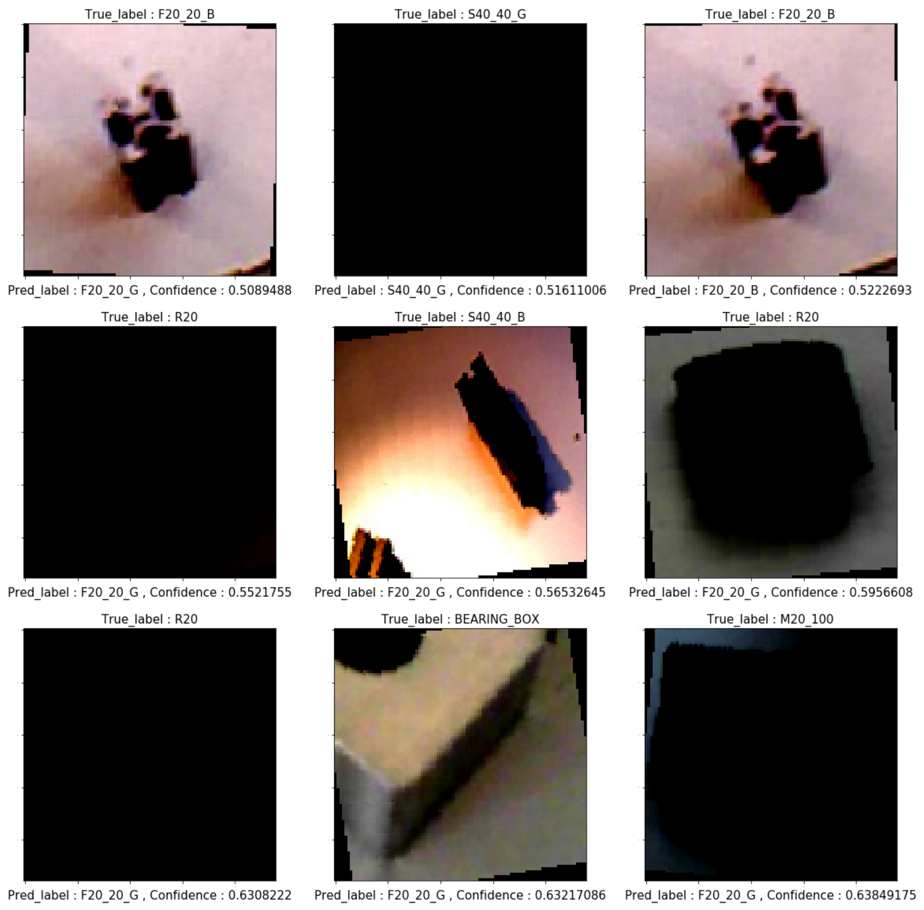
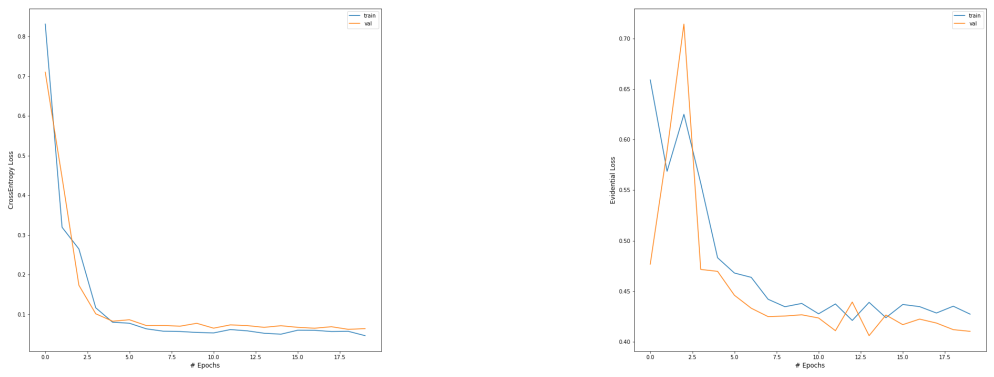

True_label : F20_20_B
Pred_label : F20_20_G , Confidence : 0.5089488
True_label : S40_40_G
Pred_label : S40_40_G , Confidence : 0.51611006
True_label : F20_20_B
Pred_label : F20_20_B , Confidence : 0.5222693
True_label : R20
Pred_label : F20_20_G , Confidence : 0.5521755
True_label : S40_40_B
Pred_label : F20_20_G , Confidence : 0.56532645
True_label : R20
Pred_label : F20_20_G , Confidence : 0.5956608
True_label : R20
Pred_label : F20_20_G , Confidence : 0.6308222
True_label : BEARING_BOX
Pred_label : F20_20_G , Confidence : 0.63217086
True_label : M20_100
Pred_label : F20_20_G , Confidence : 0.63849175

Figure 5.11: Worst predictions with evidential loss

5.2.3.2 Testing with expected value synthetic dataset

The loss of the model during training and validation with both CrossEntropy loss and EvidentialLoss is depicted in the figure 5.12a and 5.12b .



(a) Loss comparison for cross entropy

(b) Loss comparison for evidentaial

Figure 5.12: Training and validation losses for expected value synthetic dataset

The confusion matrix for the expected value synthetic test data using crossentropy model is shown in the figure 5.13 and with evidential loss is shown in the figure 5.14

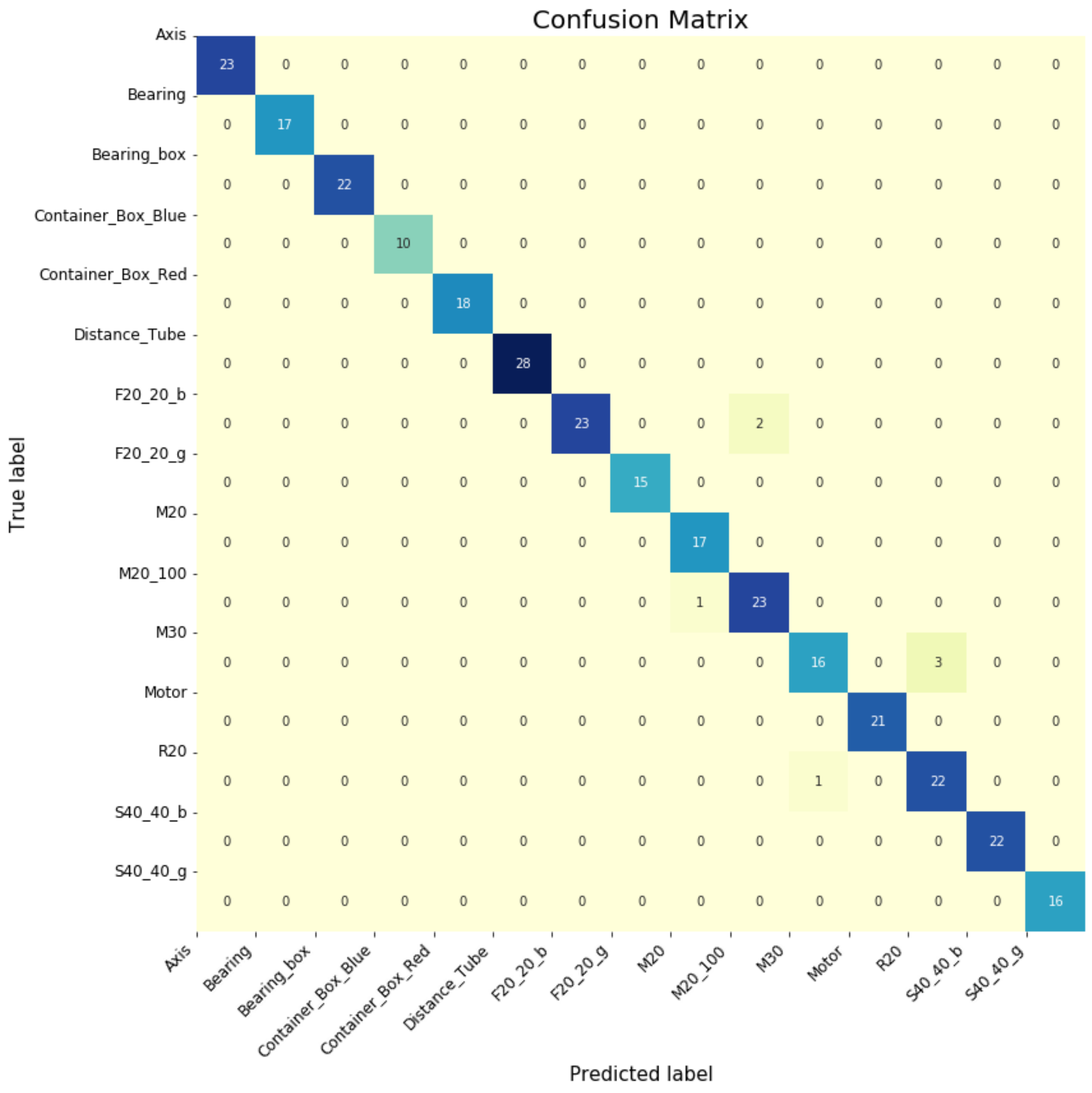
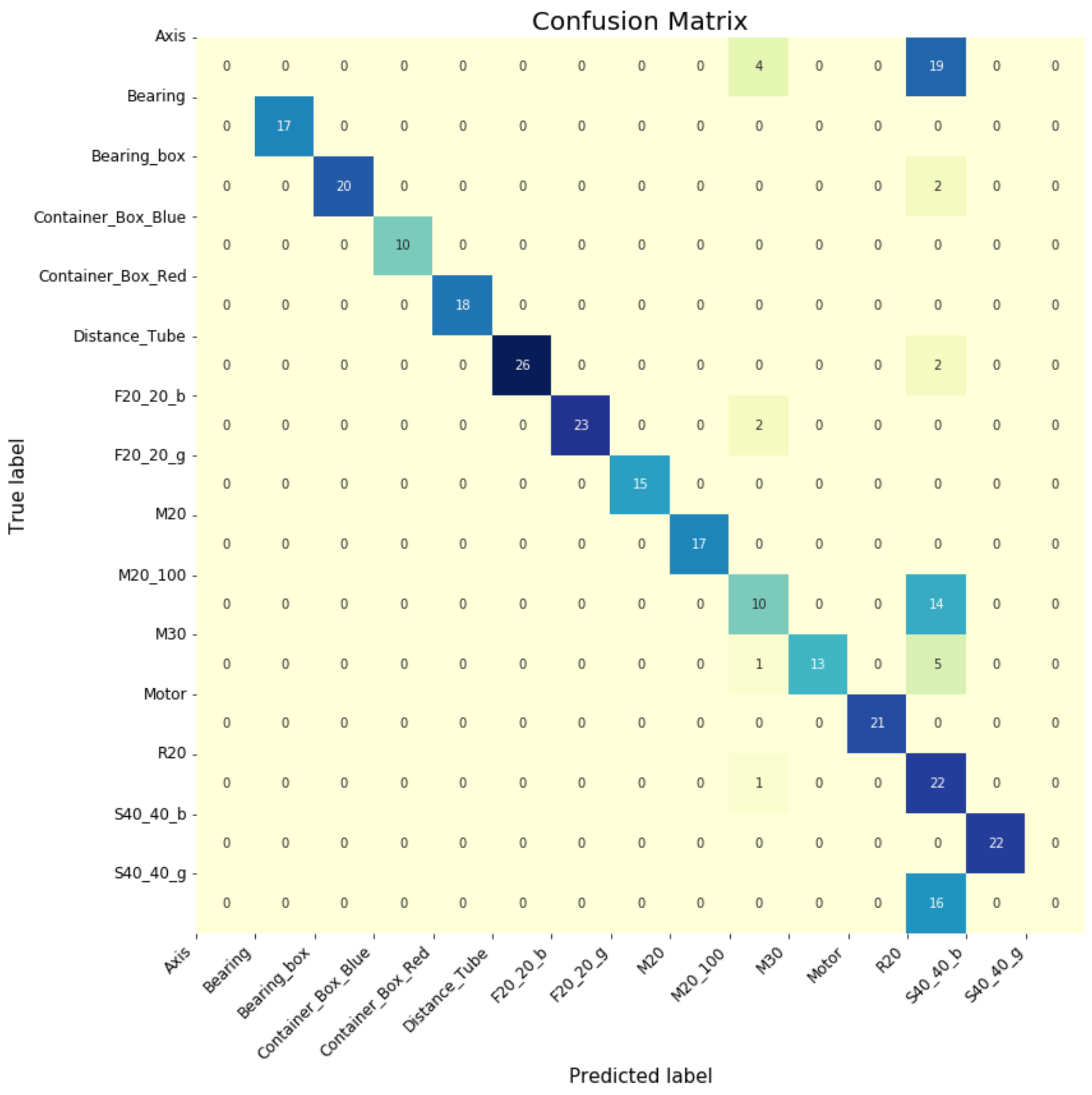


Figure 5.13: Confusion matrix of the expected value synthetic test data with cross entropy loss



	Axis	Bearing	Bearing_box	Container_Box_Blue	Container_Box_Red	Distance_Tube	F20_20_b	F20_20_g	M20	M20_100	M30	Motor	R20	S40_40_b	S40_40_g
Axis	0	0	0	0	0	0	0	0	0	4	0	0	19	0	0
Bearing	0	17	0	0	0	0	0	0	0	0	0	0	0	0	0
Bearing_box	0	0	20	0	0	0	0	0	0	0	0	0	2	0	0
Container_Box_Blue	0	0	0	10	0	0	0	0	0	0	0	0	0	0	0
Container_Box_Red	0	0	0	0	18	0	0	0	0	0	0	0	0	0	0
Distance_Tube	0	0	0	0	0	26	0	0	0	0	0	0	2	0	0
F20_20_b	0	0	0	0	0	0	23	0	0	2	0	0	0	0	0
F20_20_g	0	0	0	0	0	0	0	15	0	0	0	0	0	0	0
M20	0	0	0	0	0	0	0	0	17	0	0	0	0	0	0
M20_100	0	0	0	0	0	0	0	0	0	10	0	0	14	0	0
M30	0	0	0	0	0	0	0	0	0	1	13	0	5	0	0
Motor	0	0	0	0	0	0	0	0	0	0	0	21	0	0	0
R20	0	0	0	0	0	0	0	0	0	1	0	0	22	0	0
S40_40_b	0	0	0	0	0	0	0	0	0	0	0	0	0	22	0
S40_40_g	0	0	0	0	0	0	0	0	0	0	0	0	16	0	0

Figure 5.14: Confusion matrix of the expected value synthetic test data with evidential loss

The most confidant and least confidant predictions of the model with cross entropy loss and evidential loss are depicted in the figures 5.15 , 5.16 , 5.17 , 5.18 .

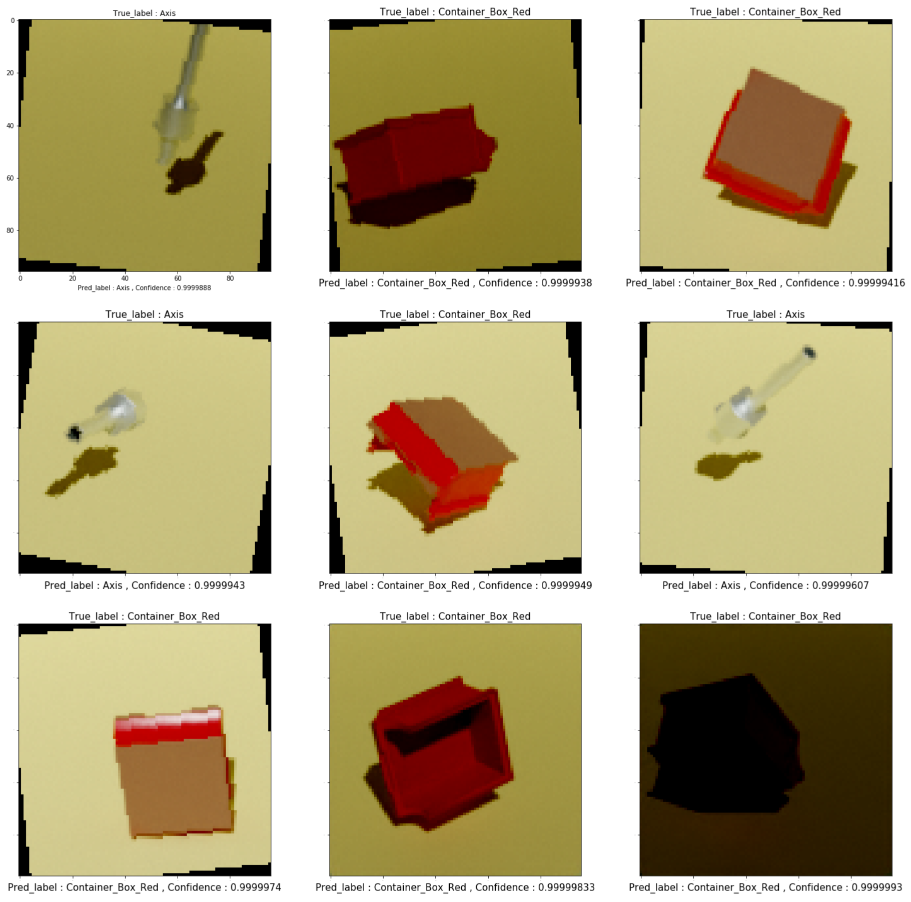

True_label : Axis
0
20
40
60
80
0
20
40
60
80
Pred_label : Axis , Confidence : 0.9999888
True_label : Container_Box_Red
Pred_label : Container_Box_Red , Confidence : 0.9999938
True_label : Container_Box_Red
Pred_label : Container_Box_Red , Confidence : 0.99999416
True_label : Axis
Pred_label : Axis , Confidence : 0.9999943
True_label : Container_Box_Red
Pred_label : Container_Box_Red , Confidence : 0.9999949
True_label : Axis
Pred_label : Axis , Confidence : 0.99999607
True_label : Container_Box_Red
Pred_label : Container_Box_Red , Confidence : 0.9999974
True_label : Container_Box_Red
Pred_label : Container_Box_Red , Confidence : 0.99999833
True_label : Container_Box_Red
Pred_label : Container_Box_Red , Confidence : 0.9999993

Figure 5.15: Best predictions with cross entropy loss

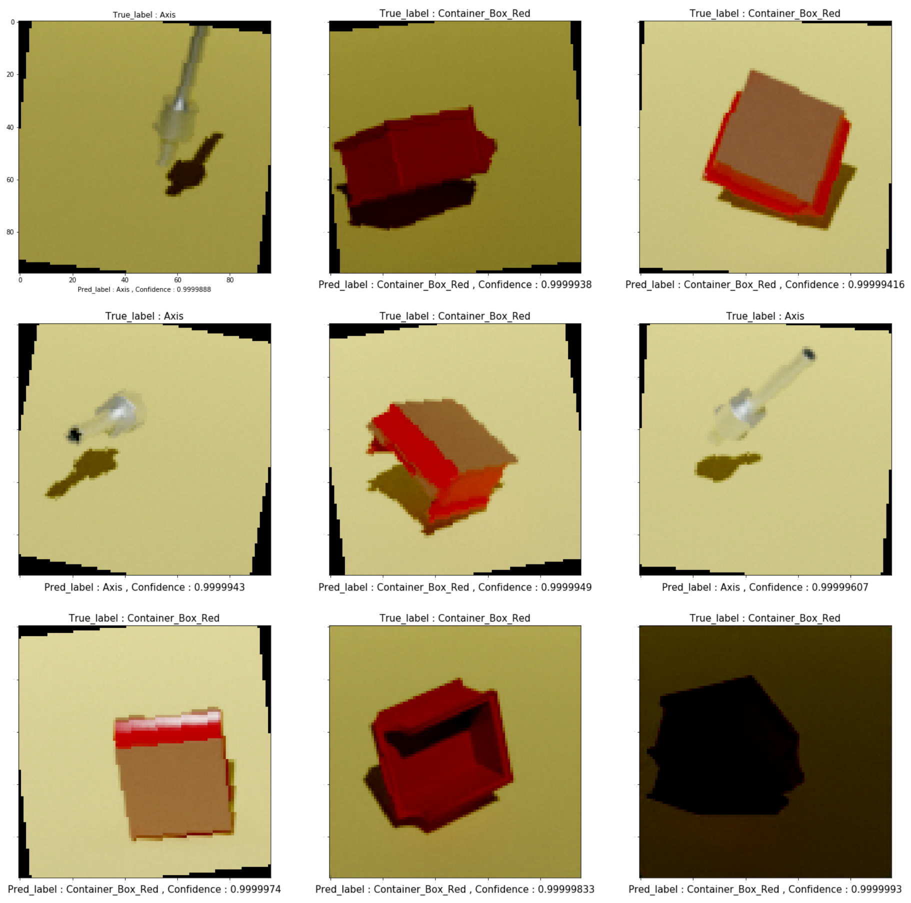

True_label : Axis
0
20
40
60
80
0
20
40
60
80
Pred_label : Axis , Confidence : 0.9999888
True_label : Container_Box_Red
Pred_label : Container_Box_Red , Confidence : 0.9999938
True_label : Container_Box_Red
Pred_label : Container_Box_Red , Confidence : 0.99999416
True_label : Axis
Pred_label : Axis , Confidence : 0.9999943
True_label : Container_Box_Red
Pred_label : Container_Box_Red , Confidence : 0.9999949
True_label : Axis
Pred_label : Axis , Confidence : 0.99999607
True_label : Container_Box_Red
Pred_label : Container_Box_Red , Confidence : 0.9999974
True_label : Container_Box_Red
Pred_label : Container_Box_Red , Confidence : 0.99999833
True_label : Container_Box_Red
Pred_label : Container_Box_Red , Confidence : 0.9999993

Figure 5.16: Worst predictions with cross entropy loss

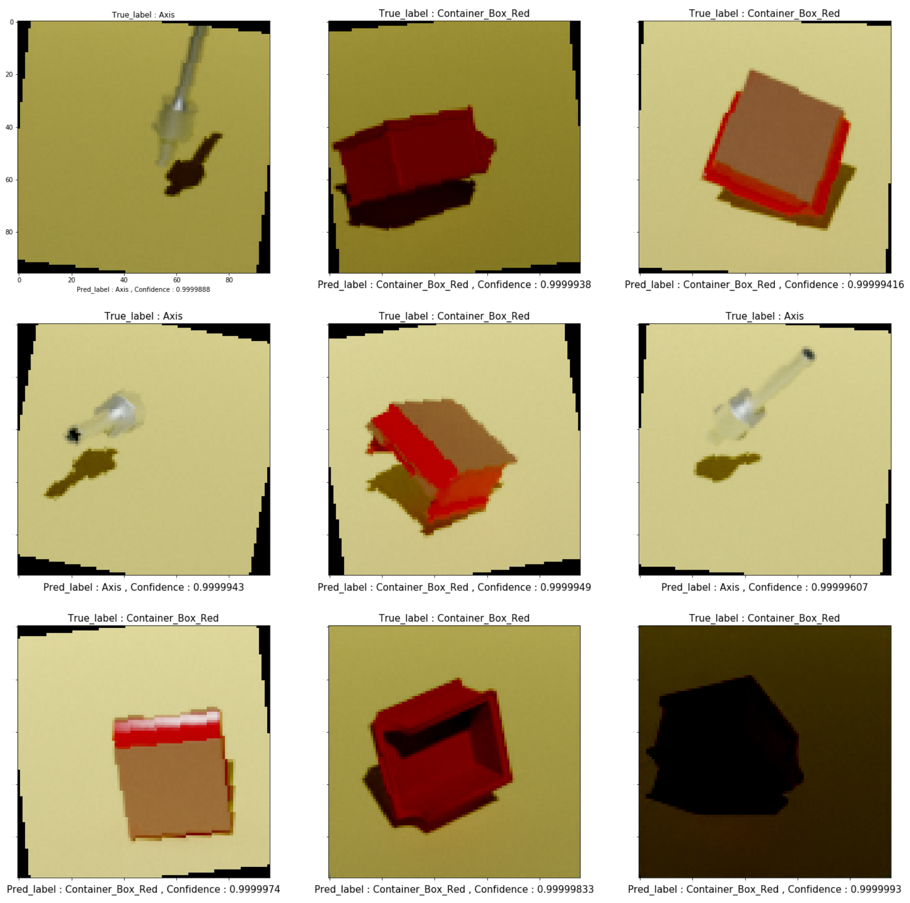

True_label : Axis
Pred_label : Axis , Confidence : 0.9999888
True_label : Container_Box_Red
Pred_label : Container_Box_Red , Confidence : 0.9999938
True_label : Container_Box_Red
Pred_label : Container_Box_Red , Confidence : 0.99999416
True_label : Axis
Pred_label : Axis , Confidence : 0.9999943
True_label : Container_Box_Red
Pred_label : Container_Box_Red , Confidence : 0.9999949
True_label : Axis
Pred_label : Axis , Confidence : 0.99999607
True_label : Container_Box_Red
Pred_label : Container_Box_Red , Confidence : 0.9999974
True_label : Container_Box_Red
Pred_label : Container_Box_Red , Confidence : 0.99999833
True_label : Container_Box_Red
Pred_label : Container_Box_Red , Confidence : 0.9999993

Figure 5.17: Best predictions with evidential loss

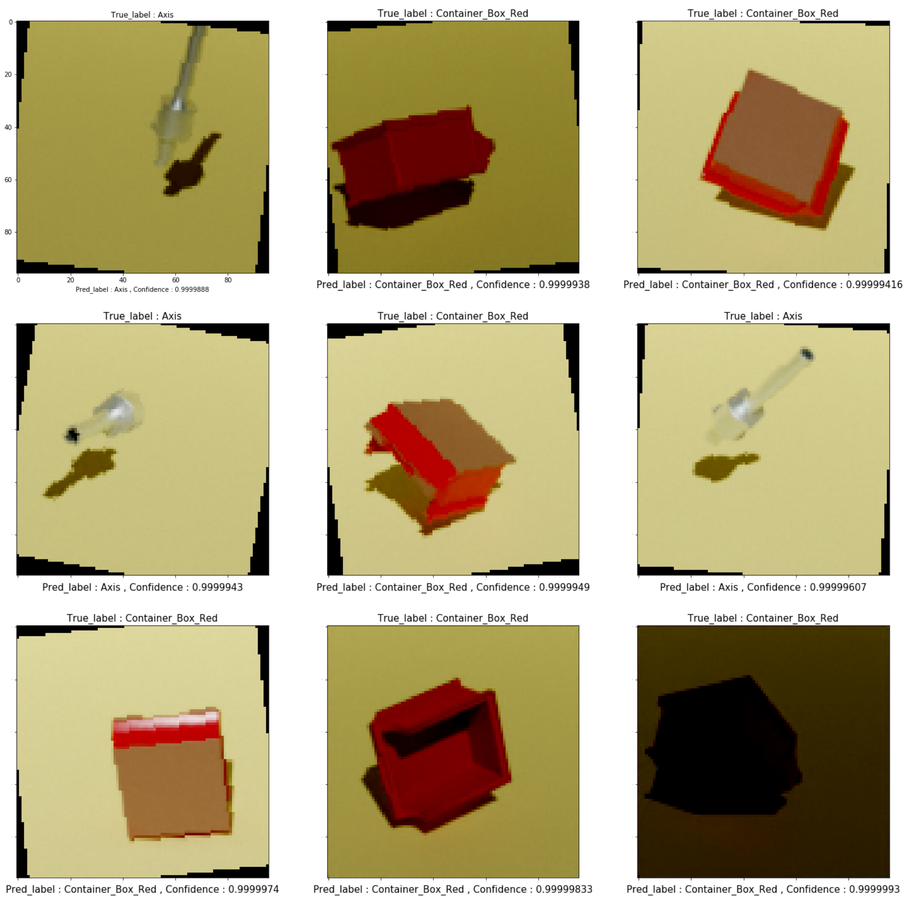
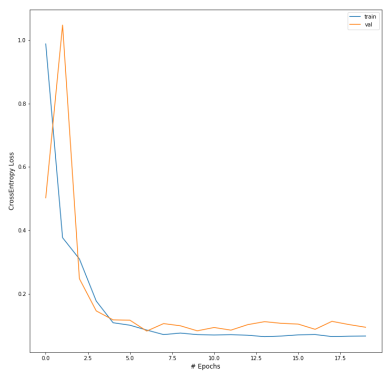

True_label : Axis
0
20
40
60
80
Pred_label : Axis , Confidence : 0.9999888
True_label : Container_Box_Red
Pred_label : Container_Box_Red , Confidence : 0.9999938
True_label : Container_Box_Red
Pred_label : Container_Box_Red , Confidence : 0.99999416
True_label : Axis
Pred_label : Axis , Confidence : 0.9999943
True_label : Container_Box_Red
Pred_label : Container_Box_Red , Confidence : 0.9999949
True_label : Axis
Pred_label : Axis , Confidence : 0.99999607
True_label : Container_Box_Red
Pred_label : Container_Box_Red , Confidence : 0.9999974
True_label : Container_Box_Red
Pred_label : Container_Box_Red , Confidence : 0.99999833
True_label : Container_Box_Red
Pred_label : Container_Box_Red , Confidence : 0.9999993

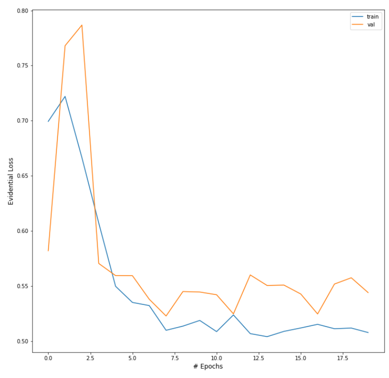
Figure 5.18: Worst predictions with evidential loss

5.2.3.3 Testing with different lighting synthetic dataset

The loss of the model during training and validation with both CrossEntropy loss and EvidentialLoss is depicted in the figure 5.19a and 5.19b



(a) Loss comparison for cross entropy



(b) Loss comparison for evidentaial

Figure 5.19: Training and validation losses for different lighting synthetic dataset

The confusion matrix for the different lighting synthetic test data generated under bright light and using crossentropy loss model is shown in the figure 5.20 and with evidential loss is shown in the figure 5.21

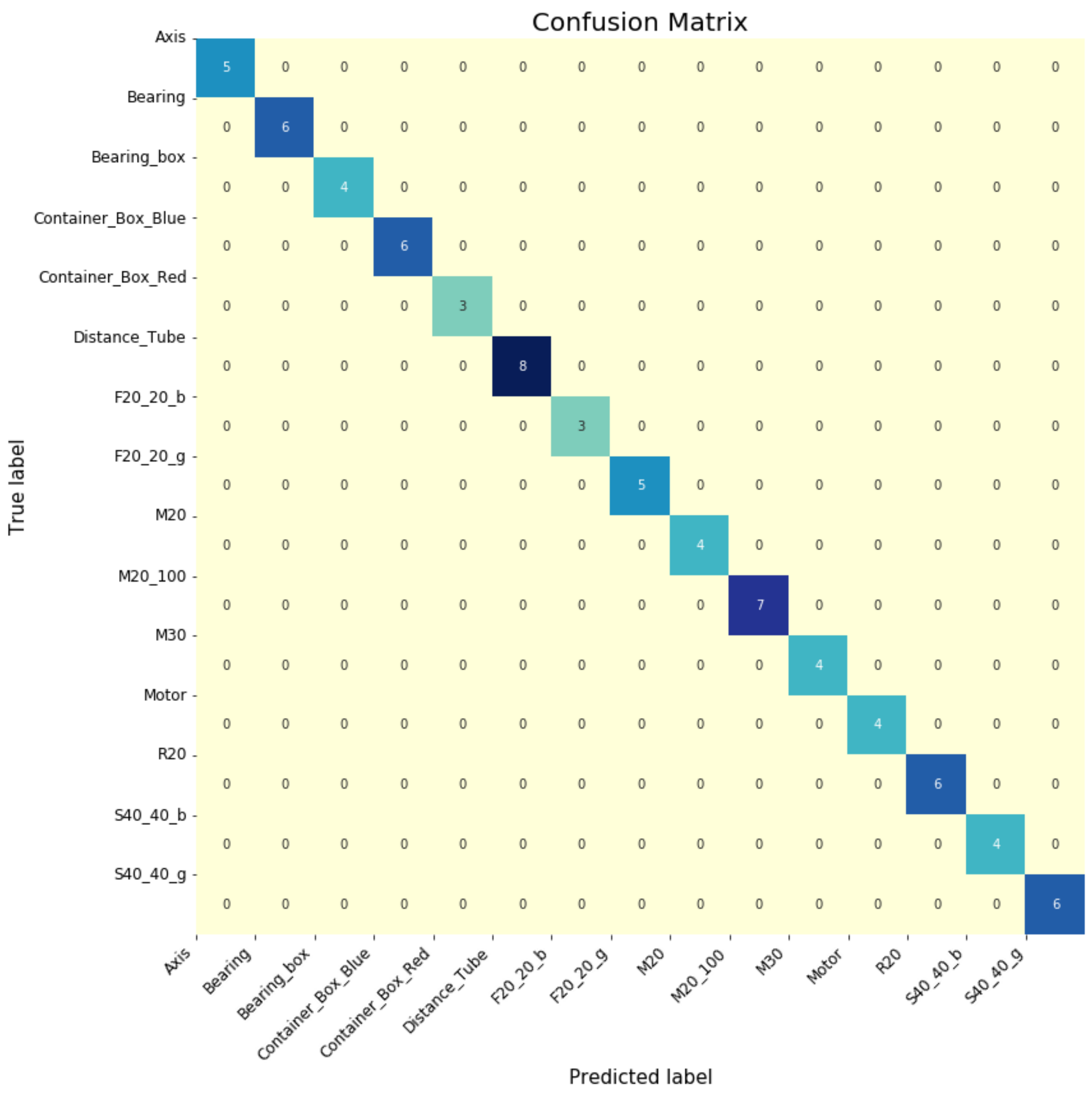
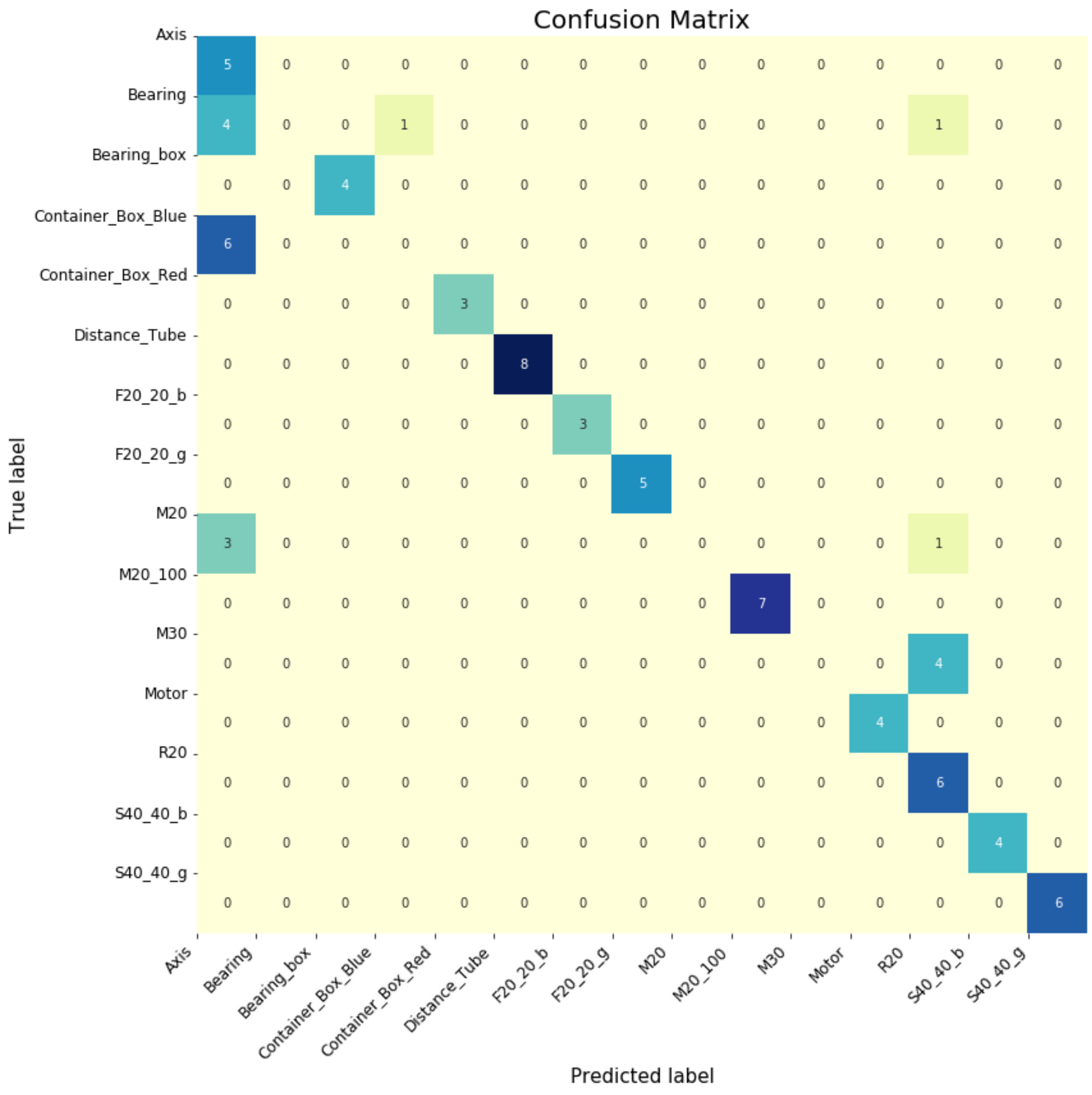


Figure 5.20: Confusion matrix of the different lighting synthetic test data under bright light with cross entropy loss



True label \ Predicted label	Axis	Bearing	Bearing_box	Container_Box_Blue	Container_Box_Red	Distance_Tube	F20_20_b	F20_20_g	M20	M20_100	M30	Motor	R20	S40_40_b	S40_40_g
Axis	5	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bearing	4	0	0	1	0	0	0	0	0	0	0	0	1	0	0
Bearing_box	0	0	4	0	0	0	0	0	0	0	0	0	0	0	0
Container_Box_Blue	6	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Container_Box_Red	0	0	0	0	3	0	0	0	0	0	0	0	0	0	0
Distance_Tube	0	0	0	0	0	8	0	0	0	0	0	0	0	0	0
F20_20_b	0	0	0	0	0	0	3	0	0	0	0	0	0	0	0
F20_20_g	0	0	0	0	0	0	0	5	0	0	0	0	0	0	0
M20	3	0	0	0	0	0	0	0	0	0	0	0	1	0	0
M20_100	0	0	0	0	0	0	0	0	0	7	0	0	0	0	0
M30	0	0	0	0	0	0	0	0	0	0	0	0	4	0	0
Motor	0	0	0	0	0	0	0	0	0	0	0	4	0	0	0
R20	0	0	0	0	0	0	0	0	0	0	0	0	6	0	0
S40_40_b	0	0	0	0	0	0	0	0	0	0	0	0	0	4	0
S40_40_g	0	0	0	0	0	0	0	0	0	0	0	0	0	0	6

Figure 5.21: Confusion matrix of the different lighting synthetic test data under bright light with evidential loss

The confusion matrix for the different lighting synthetic test data generated under dark light and using crossentropy loss model is shown in the figure 5.22 and with evidential loss is shown in the figure 5.23

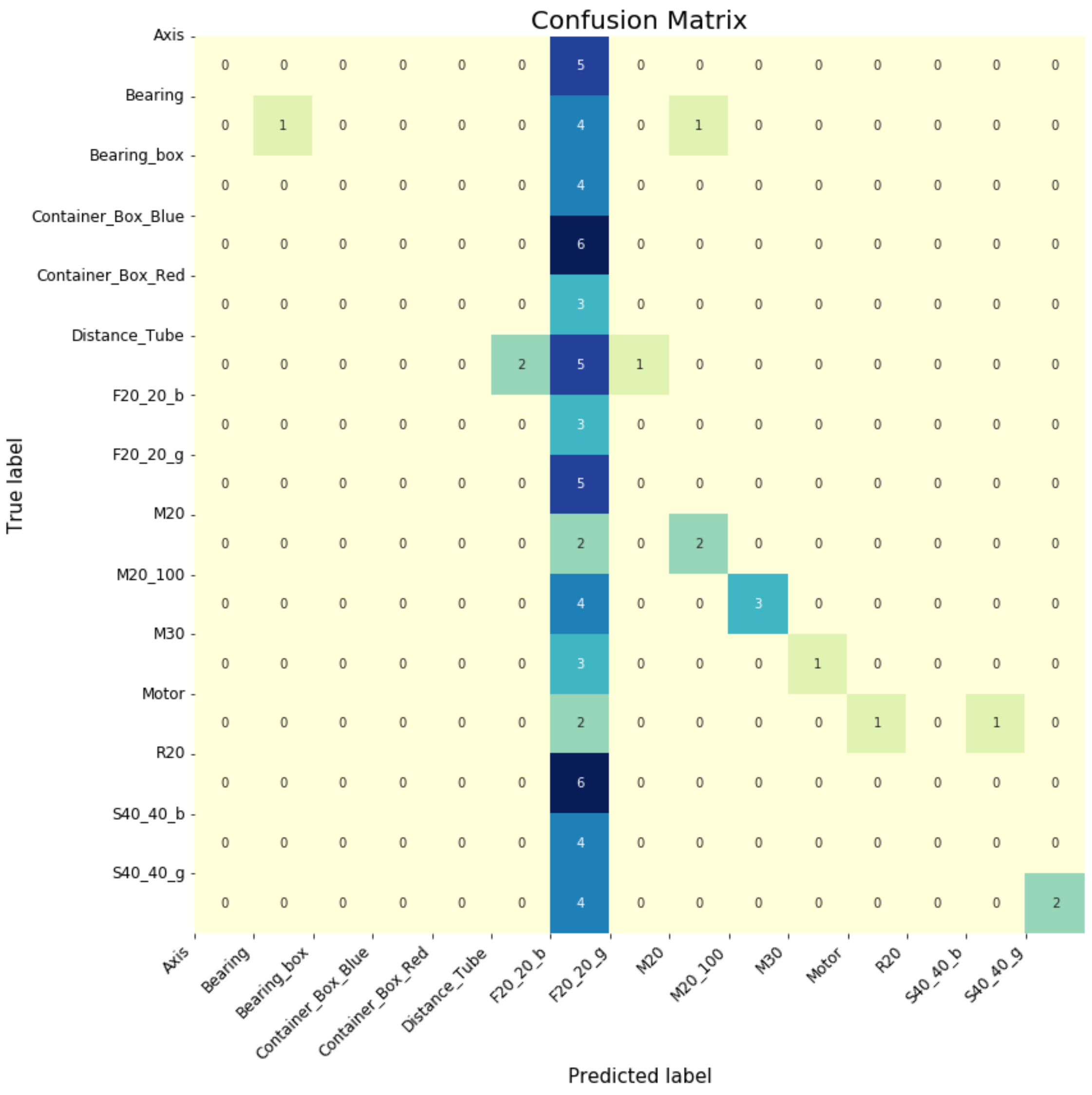
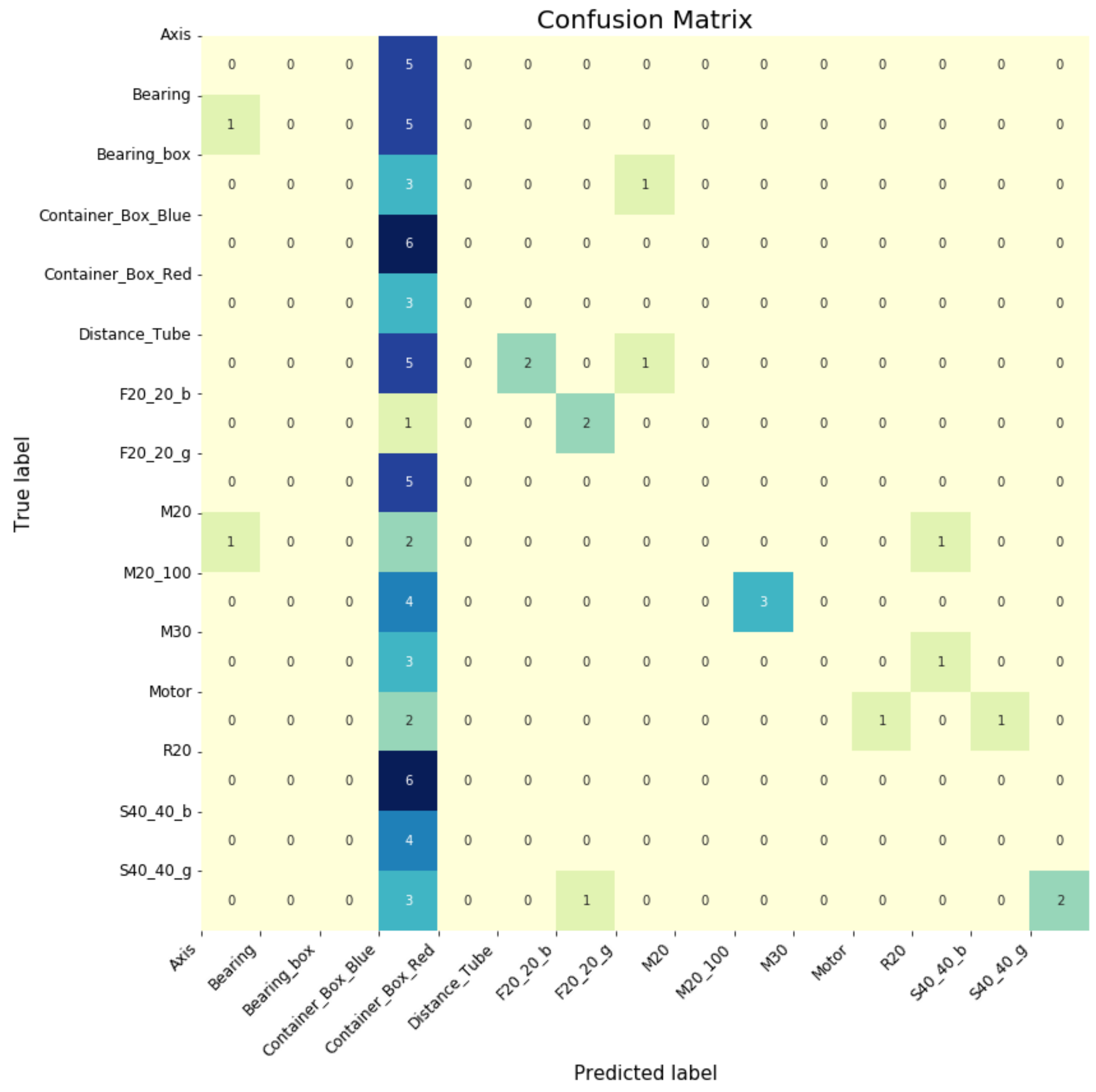


Figure 5.22: Confusion matrix of the different lighting synthetic test data under dark light with cross entropy loss



	Axis	Bearing	Bearing_box	Container_Box_Blue	Container_Box_Red	Distance_Tube	F20_20_b	F20_20_g	M20	M20_100	M30	Motor	R20	S40_40_b	S40_40_g
Axis	0	0	0	5	0	0	0	0	0	0	0	0	0	0	0
Bearing	1	0	0	5	0	0	0	0	0	0	0	0	0	0	0
Bearing_box	0	0	0	3	0	0	0	1	0	0	0	0	0	0	0
Container_Box_Blue	0	0	0	6	0	0	0	0	0	0	0	0	0	0	0
Container_Box_Red	0	0	0	3	0	0	0	0	0	0	0	0	0	0	0
Distance_Tube	0	0	0	5	0	2	0	1	0	0	0	0	0	0	0
F20_20_b	0	0	0	1	0	0	2	0	0	0	0	0	0	0	0
F20_20_g	0	0	0	5	0	0	0	0	0	0	0	0	0	0	0
M20	1	0	0	2	0	0	0	0	0	0	0	0	1	0	0
M20_100	0	0	0	4	0	0	0	0	0	3	0	0	0	0	0
M30	0	0	0	3	0	0	0	0	0	0	0	0	1	0	0
Motor	0	0	0	2	0	0	0	0	0	0	0	1	0	1	0
R20	0	0	0	6	0	0	0	0	0	0	0	0	0	0	0
S40_40_b	0	0	0	4	0	0	0	0	0	0	0	0	0	0	0
S40_40_g	0	0	0	3	0	0	1	0	0	0	0	0	0	0	2

Figure 5.23: Confusion matrix of the different lighting synthetic test data under dark light with evidential loss

The most confidant and least confidant predictions of the model with cross entropy loss are depicted in the below figures.

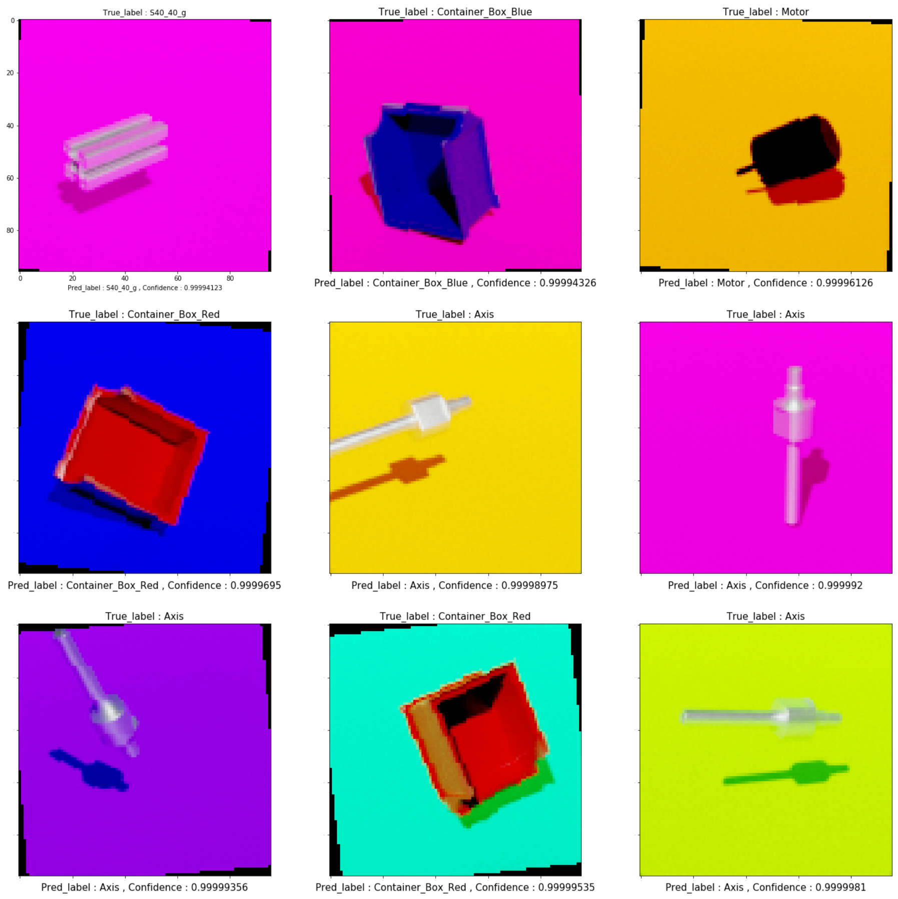

True_label : S40_40_g
0
20
40
60
80
0
20
40
60
80
Pred_label : S40_40_g , Confidence : 0.99994123
True_label : Container_Box_Blue
Pred_label : Container_Box_Blue , Confidence : 0.99994326
True_label : Motor
Pred_label : Motor , Confidence : 0.99996126
True_label : Container_Box_Red
Pred_label : Container_Box_Red , Confidence : 0.9999695
True_label : Axis
Pred_label : Axis , Confidence : 0.99998975
True_label : Axis
Pred_label : Axis , Confidence : 0.999992
True_label : Axis
Pred_label : Axis , Confidence : 0.99999356
True_label : Container_Box_Red
Pred_label : Container_Box_Red , Confidence : 0.99999535
True_label : Axis
Pred_label : Axis , Confidence : 0.9999981

Figure 5.24: Best predictions with cross entropy loss on bright light

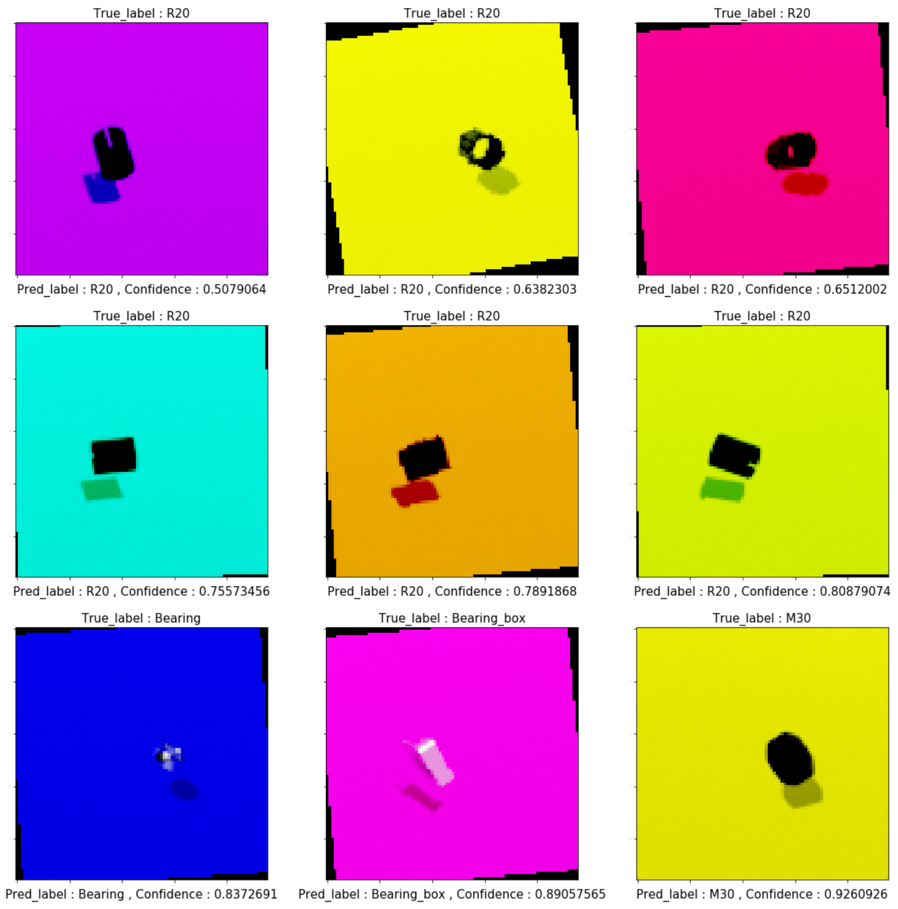

True_label : R20
Pred_label : R20 , Confidence : 0.5079064
True_label : R20
Pred_label : R20 , Confidence : 0.6382303
True_label : R20
Pred_label : R20 , Confidence : 0.6512002
True_label : R20
Pred_label : R20 , Confidence : 0.75573456
True_label : R20
Pred_label : R20 , Confidence : 0.7891868
True_label : R20
Pred_label : R20 , Confidence : 0.80879074
True_label : Bearing
Pred_label : Bearing , Confidence : 0.8372691
True_label : Bearing_box
Pred_label : Bearing_box , Confidence : 0.89057565
True_label : M30
Pred_label : M30 , Confidence : 0.9260926

Figure 5.25: Worst predictions with cross entropy loss on bright light

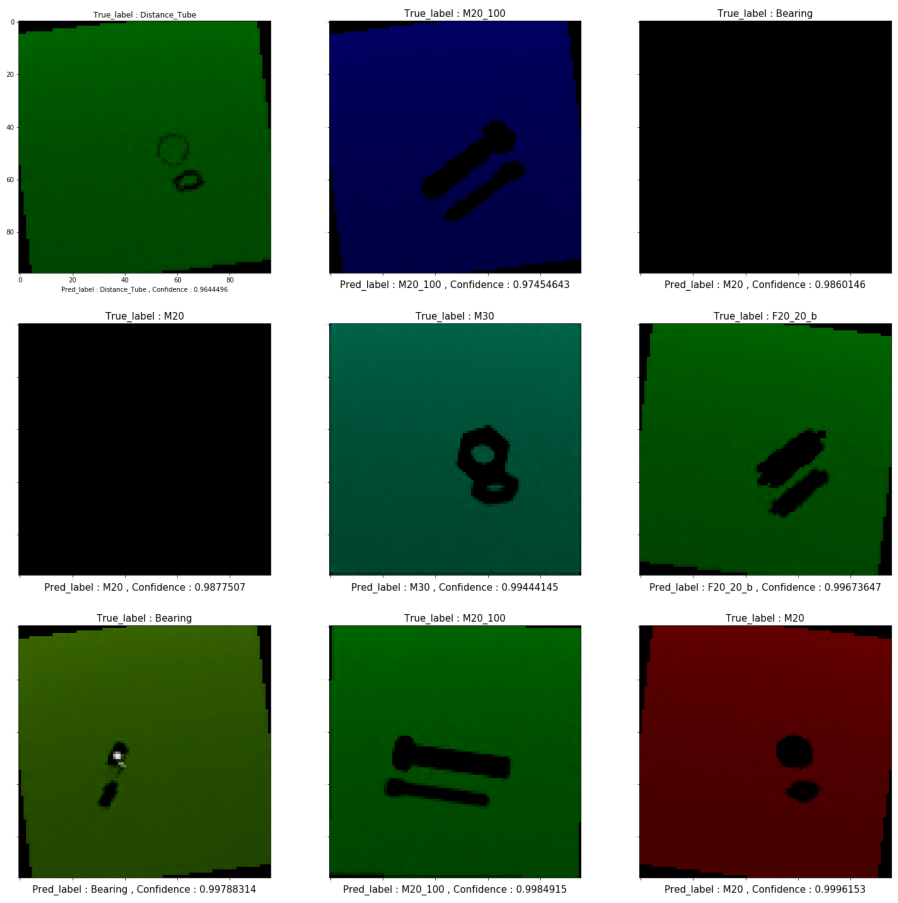

True_label : Distance_Tube
Pred_label : Distance_Tube , Confidence : 0.9644496
True_label : M20_100
Pred_label : M20_100 , Confidence : 0.97454643
True_label : Bearing
Pred_label : M20 , Confidence : 0.9860146
True_label : M20
Pred_label : M20 , Confidence : 0.9877507
True_label : M30
Pred_label : M30 , Confidence : 0.99444145
True_label : F20_20_b
Pred_label : F20_20_b , Confidence : 0.99673647
True_label : Bearing
Pred_label : Bearing , Confidence : 0.99788314
True_label : M20_100
Pred_label : M20_100 , Confidence : 0.9984915
True_label : M20
Pred_label : M20 , Confidence : 0.9996153

Figure 5.26: Best predictions with cross entropy loss on dark light

True_label : Bearing_box
Pred_label : F20_20_b , Confidence : 0.23747936
True_label : Distance_Tube
Pred_label : F20_20_g , Confidence : 0.32888654
True_label : Distance_Tube
Pred_label : Distance_Tube , Confidence : 0.45411757
True_label : Bearing
Pred_label : F20_20_b , Confidence : 0.5484906
True_label : M20_100
Pred_label : F20_20_b , Confidence : 0.5491091
True_label : R20
Pred_label : F20_20_b , Confidence : 0.55325335
True_label : M30
Pred_label : F20_20_b , Confidence : 0.5538498
True_label : Distance_Tube
Pred_label : F20_20_b , Confidence : 0.5546298
True_label : S40_40_g
Pred_label : F20_20_b , Confidence : 0.55546564

Figure 5.27: Worst predictions with cross entropy loss on dark light

The most confidant and least confidant predictions of the model with evidential loss are depicted in the below figures.

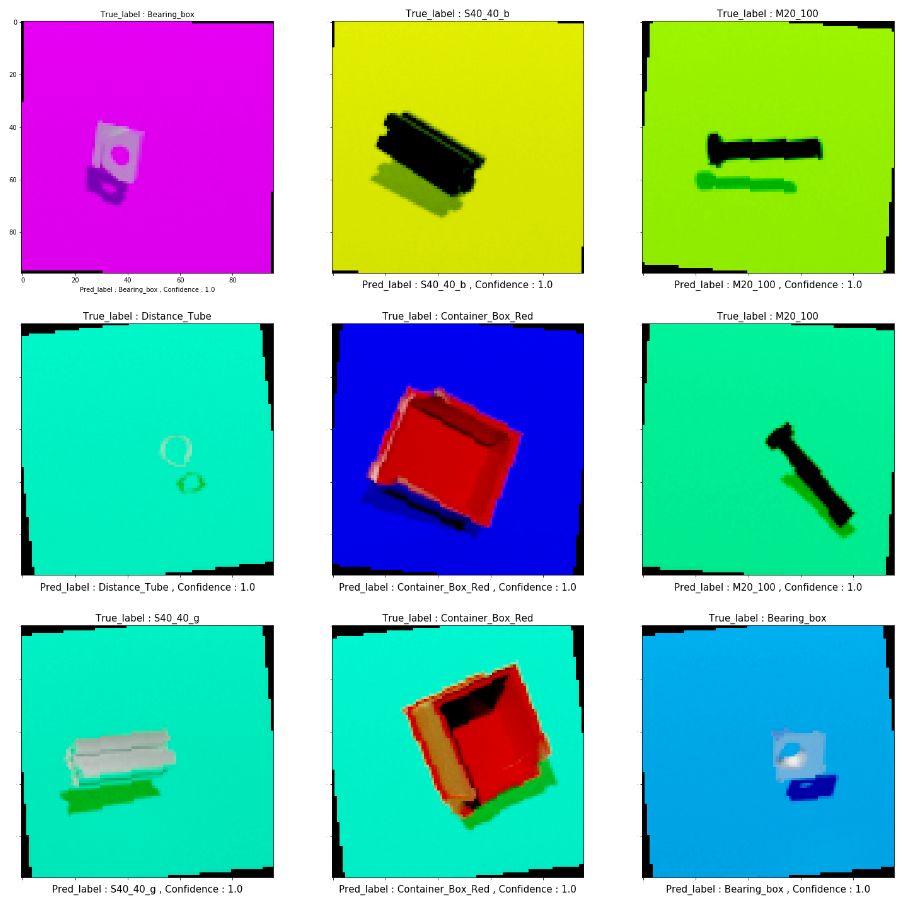

True_label : Bearing_box
Pred_label : Bearing_box , Confidence : 1.0
True_label : S40_40_b
Pred_label : S40_40_b , Confidence : 1.0
True_label : M20_100
Pred_label : M20_100 , Confidence : 1.0
True_label : Distance_Tube
Pred_label : Distance_Tube , Confidence : 1.0
True_label : Container_Box_Red
Pred_label : Container_Box_Red , Confidence : 1.0
True_label : M20_100
Pred_label : M20_100 , Confidence : 1.0
True_label : S40_40_g
Pred_label : S40_40_g , Confidence : 1.0
True_label : Container_Box_Red
Pred_label : Container_Box_Red , Confidence : 1.0
True_label : Bearing_box
Pred_label : Bearing_box , Confidence : 1.0

Figure 5.28: Best predictions with evidential loss on bright light

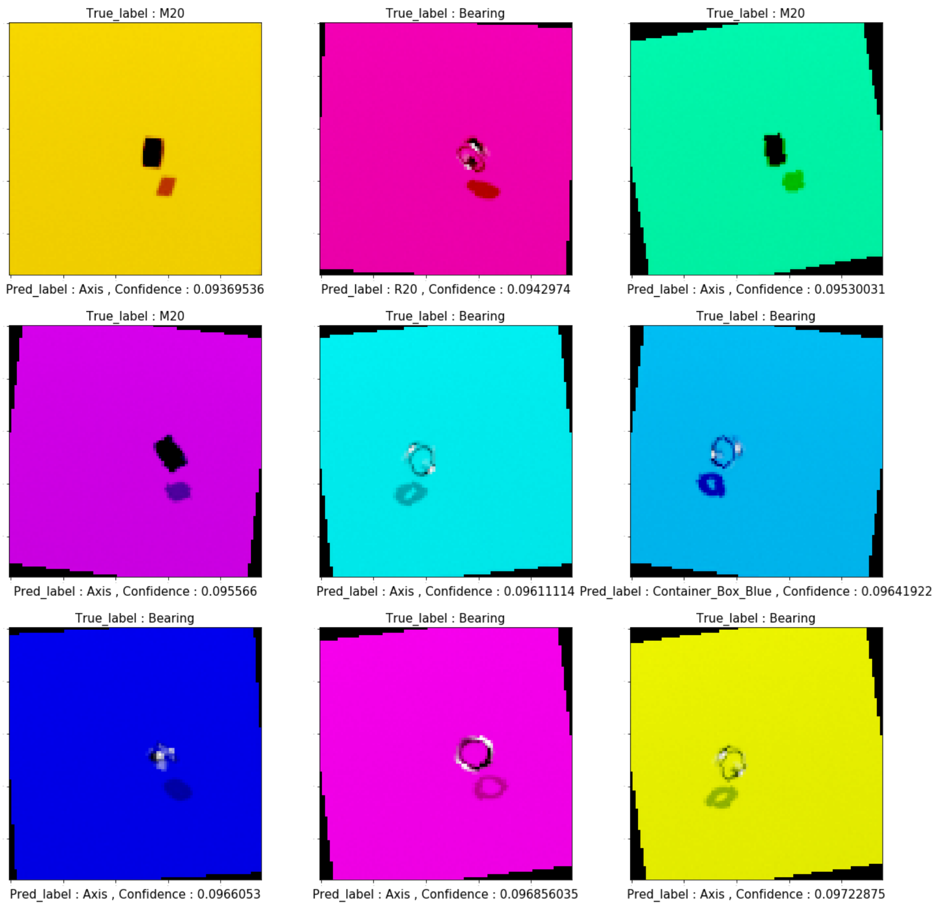

True_label : M20
Pred_label : Axis , Confidence : 0.09369536
True_label : Bearing
Pred_label : R20 , Confidence : 0.0942974
True_label : M20
Pred_label : Axis , Confidence : 0.09530031
True_label : M20
Pred_label : Axis , Confidence : 0.095566
True_label : Bearing
Pred_label : Axis , Confidence : 0.09611114
True_label : Bearing
Pred_label : Container_Box_Blue , Confidence : 0.09641922
True_label : Bearing
Pred_label : Axis , Confidence : 0.0966053
True_label : Bearing
Pred_label : Axis , Confidence : 0.096856035
True_label : Bearing
Pred_label : Axis , Confidence : 0.09722875

Figure 5.29: Worst predictions with evidential loss on bright light

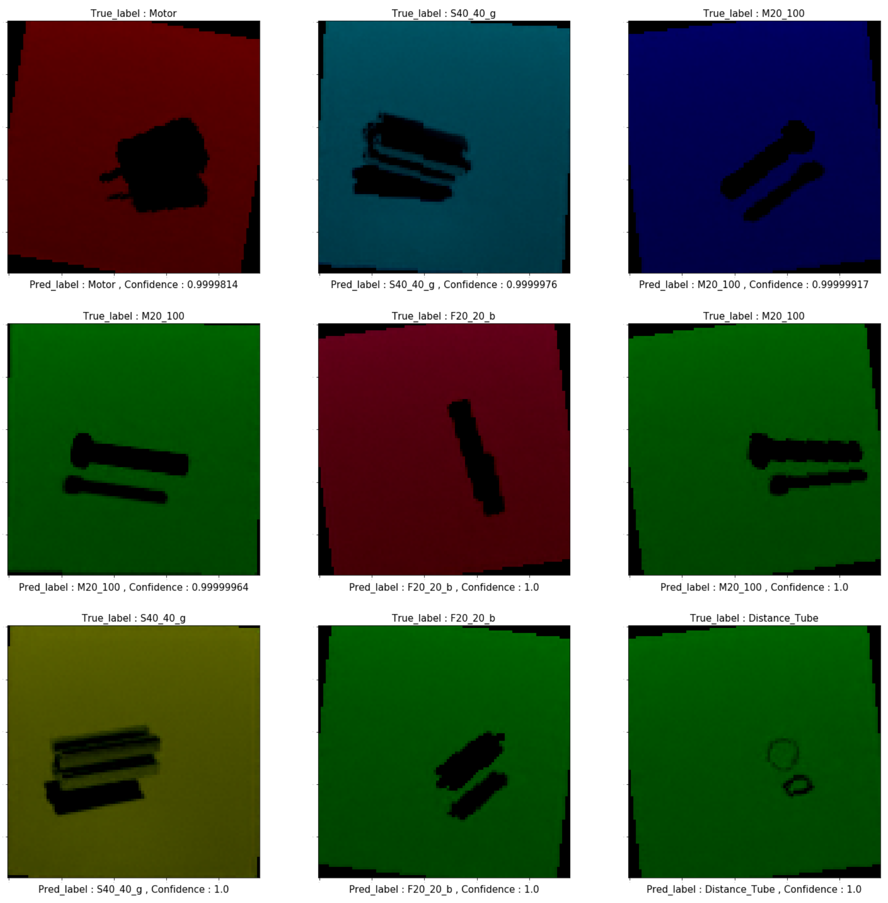

True_label : Motor
Pred_label : Motor , Confidence : 0.9999814
True_label : S40_40_g
Pred_label : S40_40_g , Confidence : 0.9999976
True_label : M20_100
Pred_label : M20_100 , Confidence : 0.99999917
True_label : M20_100
Pred_label : M20_100 , Confidence : 0.99999964
True_label : F20_20_b
Pred_label : F20_20_b , Confidence : 1.0
True_label : M20_100
Pred_label : M20_100 , Confidence : 1.0
True_label : S40_40_g
Pred_label : S40_40_g , Confidence : 1.0
True_label : F20_20_b
Pred_label : F20_20_b , Confidence : 1.0
True_label : Distance_Tube
Pred_label : Distance_Tube , Confidence : 1.0

Figure 5.30: Best predictions with evidential loss on dark light

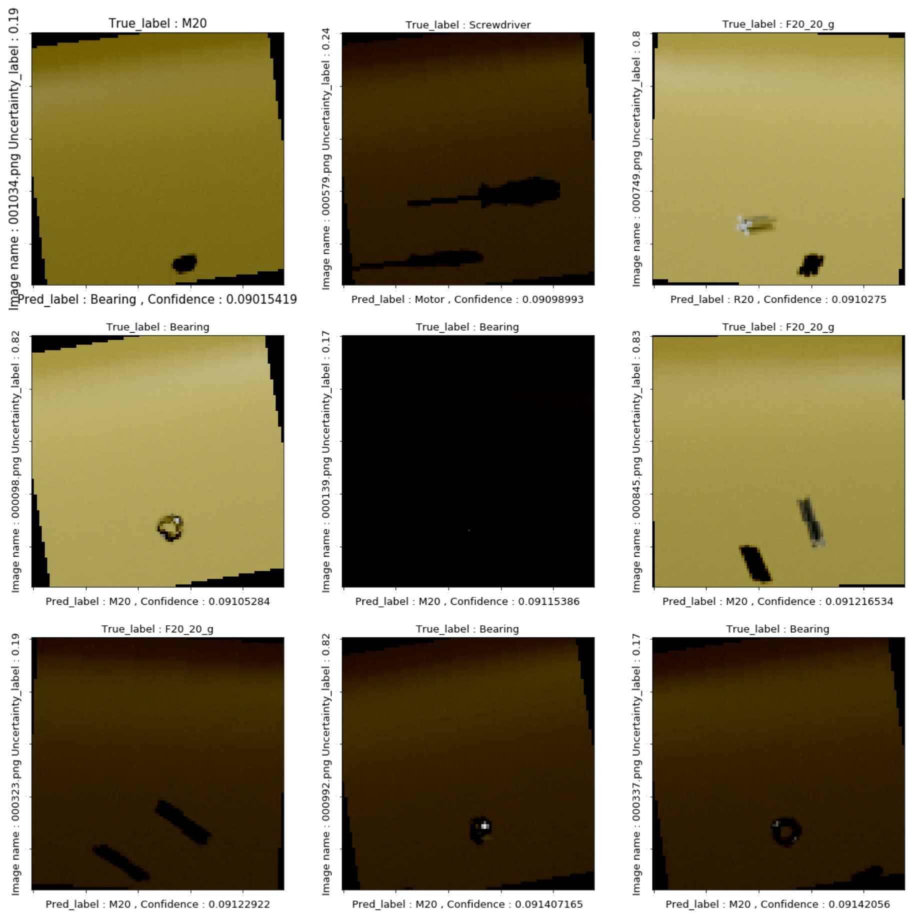

True_label : M20
Image name : 001034.png Uncertainty_label : 0.19
Pred_label : Bearing , Confidence : 0.09015419
True_label : Screwdriver
Image name : 000579.png Uncertainty_label : 0.24
Pred_label : Motor , Confidence : 0.09098993
True_label : F20_20_g
Image name : 000749.png Uncertainty_label : 0.8
Pred_label : R20 , Confidence : 0.0910275
True_label : Bearing
Image name : 000098.png Uncertainty_label : 0.82
Pred_label : M20 , Confidence : 0.09105284
True_label : Bearing
Image name : 000139.png Uncertainty_label : 0.17
Pred_label : M20 , Confidence : 0.09115386
True_label : F20_20_g
Image name : 000845.png Uncertainty_label : 0.83
Pred_label : M20 , Confidence : 0.091216534
True_label : F20_20_g
Image name : 000323.png Uncertainty_label : 0.19
Pred_label : M20 , Confidence : 0.09122922
True_label : Bearing
Image name : 000992.png Uncertainty_label : 0.82
Pred_label : M20 , Confidence : 0.091407165
True_label : Bearing
Image name : 000337.png Uncertainty_label : 0.17
Pred_label : M20 , Confidence : 0.09142056

Figure 5.31: Worst predictions with evidential loss on dark light

5.3 Results

The results of all the experiments are evaluated using the classification metrics and a comparison of the results is shown in the table 5.2 .

Datasets	Loss function	Accuracy Score	Precision Score	Recall Score	F1 Score
Real world dataset	CrossEntropy Loss	0.998	0.998	0.998	0.998
	Evidential Loss	0.991	0.992	0.991	0.992
Expected value dataset	CrossEntropy Loss	0.977	0.978	0.977	0.977
	Evidential Loss	0.78	0.779	0.78	0.762
Different lighting dataset (Bright)	CrossEntropy Loss	1.0	1.0	1.0	1.0
	Evidential Loss	0.733	0.645	0.733	0.669
Different lighting dataset (Dark)	CrossEntropy Loss	0.2	0.504	0.2	0.238
	Evidential Loss	0.213	0.368	0.213	0.202

Table 5.2: Classification results of the performed experiments

5.3.1 Limitations

The proposed approach of generating expected value dataset has the following limitations.

- The threshold for light value has to be changed based on the material of the object. This is because different materials produce different reflections for the same light value while rendering the images.
- Defining the threshold value of light for each material is challenging.
- The approach uses Fuzzy logic to calculate the expected uncertainty value.The concept of fuzzy logic depends up on the membership function.
- Defining the membership function based on the light data to get the uncertainty value is challenging.
- This is because if we change the membership function the value of uncertainty will also gets changed.

6 Conclusions

6.1 Contributions

dataset generation tool for classification and uncertainty labels aswell.

6.2 Lessons learned

Write about problems with fuzzy logic

6.3 Future work

Write about the confusion matrix method

A
Design Details

Your first appendix

B

Parameters

Your second chapter appendix

References

[1] A. Name, “Book title,” *Lecture Notes in Autonomous System*, vol. 1001, pp. 900–921, 2003.

[2] “How to render in blender,” https://artisticrender.com/how-to-render-in-blender/#:~:text=To%20render%20in%20Blender%20Press,or%20render%20animation%20from%20there., aRTISTIC RENDERER 3d art workflow.