Pig Pose Estimation Based on Extracted Data of Mask R-CNN with VGG Neural Network for Classifications

Sang Kwan Lee
South Dakota State University

Follow this and additional works at: https://openprairie.sdstate.edu/etd

Part of the Computer Sciences Commons, and the Electrical and Computer Engineering Commons

Recommended Citation
Lee, Sang Kwan, "Pig Pose Estimation Based on Extracted Data of Mask R-CNN with VGG Neural Network for Classifications" (2020). Electronic Theses and Dissertations. 4098.
https://openprairie.sdstate.edu/etd/4098

This Thesis - Open Access is brought to you for free and open access by Open PRAIRIE: Open Public Research Access Institutional Repository and Information Exchange. It has been accepted for inclusion in Electronic Theses and Dissertations by an authorized administrator of Open PRAIRIE: Open Public Research Access Institutional Repository and Information Exchange. For more information, please contact michael.biondo@sdstate.edu.
PIG POSE ESTIMATION BASED ON EXTRACTED DATA OF MASK R-CNN WITH VGG NEURAL NETWORK FOR CLASSIFICATIONS

BY

SANG KWAN LEE

A thesis submitted in partial fulfillment of the requirements for the

Master of Science

Major in Computer Science

South Dakota State University

2020
THESIS ACCEPTANCE PAGE

Sang Kwan Lee

This thesis is approved as a creditable and independent investigation by a candidate for the master’s degree and is acceptable for meeting the thesis requirements for this degree. Acceptance of this does not imply that the conclusions reached by the candidate are necessarily the conclusions of the major department.

Kwanghee Won
Advisor

Date

Sid Suryanarayanan
Department Head

Date

Dean, Graduate School

Date
ACKNOWLEDGMENT

I am thankful from my thesis advisor Dr. Kwang Hee Won. Without from his help and guidance I would not be able to finish my thesis. I appreciate his advice for my future research and studies. I do appreciate instructors from the SDSU’s electrical engineering and computer science department especially to Dr. Shin, Dr. Hammer, Ms. Kurtenbach and Dr Hansen. Due to all the help from the instructors, I was lucky and grateful to study in such a helpful environment.
CONTENTS

ABBREVIATIONS ........................................................................................................... v

LIST OF FIGURES ........................................................................................................ vi

LIST OF TABLES ........................................................................................................... vii

ABSTRACT .................................................................................................................... ix

INTRODUCTION ............................................................................................................ 1

RELATED WORK .......................................................................................................... 4

Mask R-CNN .................................................................................................................. 4

Instance Segmentation ................................................................................................. 5

Region Proposal Network (RPN) .................................................................................... 6

Feature Pyramid Network (FPN) .................................................................................... 7

ROI Align ....................................................................................................................... 8

VGG-16 network ........................................................................................................... 9

Activation functions ..................................................................................................... 10

Hourglass network ....................................................................................................... 12

PROBLEMS .................................................................................................................... 13

NETWORK DESIGN ....................................................................................................... 19

EXPERIMENT RESULT ............................................................................................... 25

CONCLUSION ............................................................................................................... 29

REFERENCES .............................................................................................................. 31
<table>
<thead>
<tr>
<th>Abbreviation</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mask R-CNN</td>
<td>Mask Region based Convolutional Neural Network</td>
</tr>
<tr>
<td>VGG</td>
<td>Visual Geometry Group</td>
</tr>
<tr>
<td>NN</td>
<td>Neural Network</td>
</tr>
<tr>
<td>RPN</td>
<td>Region Proposal Network</td>
</tr>
<tr>
<td>FPN</td>
<td>Feature Pyramid Network</td>
</tr>
<tr>
<td>ROI</td>
<td>Region of Interest</td>
</tr>
<tr>
<td>ROI Pool</td>
<td>Region of Interest Pooling</td>
</tr>
<tr>
<td>ROI Align</td>
<td>Region of Interest Align</td>
</tr>
<tr>
<td>Relu</td>
<td>Rectified Linear Unit</td>
</tr>
<tr>
<td>cls</td>
<td>Class</td>
</tr>
</tbody>
</table>
LIST OF FIGURES

Figure 1. Mask R-CNN Structure [4] .................................................................5

Figure 2. Instance segmentation results from Mask R-CNN of pig detection. ..........6

Figure 3. RPN structure [5]. ...............................................................................7

Figure 4. FPN structure. ....................................................................................8

Figure 5. ROI Align example. .............................................................................9

Figure 6. VGG-16 network structure. .................................................................10

Figure 7. Neural network layer example. ...........................................................12

Figure 8. Hourglass shape network. .................................................................13

Figure 9. Four-channel output from the input image [3]. ....................................14

Figure 10. Channel outputs. ...............................................................................16

Figure 11. Diagram converting the 16 channels into instance locations [3]. ..............17

Figure 12. Problem associating the nearest body parts utilizing Euclidian distance. ....18

Figure 13. Workflow process of the proposed approach. ....................................19

Figure 14. Example of masking augmentation. ..................................................21

Figure 15. Development of augmented mask RGB image. ....................................22

Figure 16. Example of augmented data. .............................................................23

Figure 17. Model accuracy and loss of the augmented masked RGB image training. .....26
Figure 18. Results of miss classified data based on the trained model. ..................27
LIST OF TABLES

Table 1. Paired vector table [3] ..................................................................................15
ABSTRACT

PIG POSE ESTIMATION BASED ON EXTRACTED DATA OF MASK R-CNN WITH VGG NEURAL NETWORK

SANG KWAN LEE

2020

This paper proposes a pig pose estimation operating with Region Proposal Network (RPN) of Mask Region based Convolutional Neural Network (Mask R-CNN) and Visual Geometry Group (VGG) Neural Network (NN). Object pose estimations generates from the associations of different key points. Key points could be explained as specific location of an object such as different joints of a human body or joints of different object. Hourglass network is one of a NN delivering key points of an object. Associating the different key points with the hourglass network results could be represented as instance-level detection [3]. However, the instance-level detection shows a lack of accuracy on the results. This algorithm provides limitations on the accuracy because the pairwise association is not produced on individual pigs which means extra calculation must be handled to connect the different body parts. During the process of associating the body parts, parts from other pigs might be involved.

Mask R-CNN presents a feature of Region Proposal Network (RPN) which categorizes distinct objects of an input image depending on the model trained [4]. In this paper we introduce a method providing the pig pose estimation constructed from the Mask R-CNN’s masking results. 230 images were operated as a dataset. An average of 14 pigs appeared in each of the 230 images. The VGG Neural Network was utilized for
classifying the pig standing up or laying down position with the masked RGB image extracted from the Mask R-CNN.
INTRODUCTION

There exist various type of algorithms of object detection and tracking in computer vision. Object detection would be locating the object from a given image. Tracking the object would be following each consecutive object in between various of frames. Many algorithms were introduced for different classifications or pose estimation in NN. NN can be implemented for automated tracking, classification of objects or object segmentations and more. The individual objects could be differentiated through the 3D trajectories of point clouds [1]. Other NN such as Segnet, U-net, or VGG Networks can be some of the examples for object classifications in computer visions [13, 14]. Pose estimation is generated from NN which provides specific key point locations of an object in an image and predicts the pose with the associated key points. The key points could be represented as joints of a human body, key locations of an object etc. However key pose estimation sometimes suffers from lack of accuracy when connecting the different joints together. A risk might appear within the key points connecting to other key points of distinct objects.

Hourglass network is a combination of Segnet and U-net extracting feature location of the input image. In this paper, we examine the problems discovered while running with the Hourglass network and connecting associations, to provide a different approach for pose estimation. This work introduces a method by managing the masking result of Mask R-CNN, converting it to a mask RGB image and running it through a VGG neural network for different pose estimations and classification. This approach was projected as the associations of each joints are not created through individual object but throughout the entire image. Mask R-CNN delivers a characteristic which the RPN
proposes a candidate object bound boxes and extract features using Region of Interest Pooling (RoIPool) from each candidate box [4]. This specialty of Mask R-CNN differentiates all detected objects which could provide a better answer for object distinction. Mask R-CNN generates an instance segmentation which detects and seeks the pixel values defining a specific object [4].

With the extracted data of Mask R-CNN, we utilize the VGG network with only two classification layers for pose estimation. The pig pose estimation is differentiated as the pig standing up or pig laying down. We wanted to examine the characteristic of an object through different classifications. It would be possible to investigate the different object pose through different key point associations but as mentioned earlier the association of different key points could be hard to distinguish individual objects. Several networks exist for classification purpose and VGG-16 was selected based on the overall performance of ImageNet Large-Scale Visual Recognition Challenge (ILSVRC) [7, 8, 9].

Total 230 images, each affording 14 individual pigs on average in the images were utilized as datasets into the Mask R-CNN. We extracted the ground truth mask information from the Mask R-CNN which is the targeted or the actual masking information which a NN tries to reach. From the extracted information of Mask R-CNN, a masked RGB image was produced. 15,465 of masked RGB images were generated from the Mask R-CNN. 4735 images of pigs standing up and laying down were categorized among 15,465 masked RGB images of pigs. The mask RGB image was presented based on the masking information and the Region of Interest (ROI) of the Mask R-CNN. Total 9470 of mask RGB dataset was trained into a VGG network with two
classification. The training of the VGG neural network was operated in Google Colab[20] for GPU usage purpose.
RELATED WORK

In this section, we provide the related algorithms within the thesis project. We do provide some detailed explanation of the Mask R-CNN and the key algorithms within. We do explain some information about the overall concept of a NN and its activation function for better understanding how the NN operates. We also generate some explanations about the VGG network being operated as the classification methods. We do describe how the hourglass network is managed. Based on the key algorithms of Mask R-CNN, we could offer a better understanding on how the object detections and individuality is managed. One of the main approaches of this thesis is to distinguish individual objects from RPN and masking data of the Mask R-CNN. Another approach would be classifying the characteristic of an individual pig through the VGG network.

Mask R-CNN:

Mask R-CNN is an extension of Faster R-CNN. Faster R-CNN gives an output of a class object and a bounding-box offset. Mask R-CNN adds a third branch which outputs the object mask. The first stage of Mask R-CNN generates the RPN which provides candidates of object boxes. The second stage operates with Region of Interest Align (ROI Align) which delivers classification and bounding box regression. Mask R-CNN also outputs a binary mask for each Region of Interest (ROI) in parallel with prediction of the class and box offset [4]. Overall Mask R-CNN delivers an instance segmentation from the input images. Figure 1 shows a picture of the Mask R-CNN structure.
Instance Segmentation:

Instance Segmentation is a combination of both object detection and semantic segmentation which also defines the output of Mask R-CNN. In other words, classifying the pixel information detected based on the neural network operated with. Mask R-CNN have a characteristic of predicting the class labels and segmentation in parallel [4]. The object category found is differentiated by several instances. Below shows an example of an instance segmentation based on the Mask R-CNN output.
Region Proposal Network (RPN):

RPN takes an image as an input and provides a set of object box proposals, each of them with an object score [5]. A small network over the convolutional feature map output slides with the last shared convolutional layer [5]. On each sliding window location, we predict multiple region proposals [5]. The total number of region proposals on each location is denoted by $k$. The output of RPN is generated as $2k$ numbers of $cls$ (class) layers with the score and $4k$ numbers of $reg$ layers with the coordinates of $k$ boxes [5]. The $k$ number of proposed boxes is generated as anchors. Anchors are centered at
each sliding window and associate with a scale and aspect ratio [5]. Below shows an example of the RPN.

![RPN structure](image)

**Figure 3. RPN structure [5].**

Feature Pyramid Network (FPN):

FPN focusses on the sliding window of RPN. An input image is processed through a fully convolutional network outputting several feature maps on each level. FPN is differentiate by the bottom-up pathways, top-down pathways and lateral connections [6]. The bottom-up pathway signifies the backbone of FPN and Resnet Convolutional neural network operates as the backbone of it [6]. Each output features of each layers are denoted as \{C2, C3, C4, C5\} for conv2, conv3, conv4 and conv5 [6]. Top-down pathways gathers the matching feature information from bottom-up pathways and generate a lateral connection during up-sampling [6]. Lateral connection merges the different features by addition. The up-sampling maps combines with the matching map
by element-wise addition [6]. Once merged, a 3 x 3 convolution proceeds to create the prediction or the feature maps [6]. The produced feature maps are known as \{P2, P3, P4, P5\} [6]. Below shows a picture on the structure of FPN.

![FPN structure](image)

**Figure 4. FPN structure.**

Region of Interest Align (ROI Align):

Mask R-CNN applies the RoI Align instead of Region of Interest Pooling (RoI Pool) from Faster R-CNN. A box is determined based on the ROI and the size of the pooling layer on the feature map. ROI Align computes the value of each sampling point by bilinear interpolation from the nearby grid points of the feature map [4]. ROI Align has a characteristic of not having definite pixel in grid because the coordinates are generated as float values. Each grid points can be calculated based on the pooling size and the ROI size. Based on the characteristic of RoI Align, the value of each feature represents to be more precise compared to RoI Pool. Below shows an example of ROI
Align of 2 x 2 grid. Based from figure 5, Each dot generates a value based on the Bilinear interpolation. Operate with max pooling or average pooling to gather new values for the next feature.

![Feature map](image)

Figure 5. ROI Align example.

VGG-16 network:

VGG network is a type of a fully convolutional network which operates mostly in classification purposes [8]. Depending on the layer size of the network, the number representing shows to be different. VGG-16 network has total of 16 weight layers. The network implemented for the thesis operates with the same VGG-16 network with only managing a different classification at the last layer. The activation function is calculated by Rectified Linear Unit (Relu) for each convolutional layer [8]. The last layer operates with the Softmax activation function. Below Shows the structure of the VGG network trained.
Activation functions:

Activation functions are calculation layers to compute the weighted sum of input and biases, of which to decide if a neuron will be fired or not [11]. The purpose of utilizing the activation functions is to avoid non-linearity in the neural network. The network with the activation function provides a gradient descent and an output which contains the parameter of the data [11]. Each layer executes an activation function utilizing its result as the next input of the following hidden layer. Sometimes in deep learning, problems such as vanishing gradient descent or exploding gradient descent arises during the back propagation [11]. To avoid such problems, different activation functions such as Relu or other functions are utilized. In this network model Relu and
sigmoid functions has been operated. The equations of utilized activation functions deliver as below.

Sofmax function

\[ f(x) = (1 / (1 + exp^x)) \]

Relu function

\[ f(x) = \max(0, x) \begin{cases} x_i & \text{if } x_i > 0 \\ 0 & \text{if } x_i < 0 \end{cases} \]

Softmax functions produces an output between a range of 0 and 1, with the sum of the probabilities equal to 1 [11]. Softmax function operates usually in multivariate classification which returns a probability of each class. Relu function represent a threshold operation to each input elements where value less than zero are set to zero [11]. Relu represents nearly as a linear function [11]. Figure 7 shows a diagram of a neural network. Based from figure 7, each neuron receives a sum of input and weights multiplied and process through an activation function. Need to add the bias in each neuron before generating the activation function. The bias helps to shift the activation functions to left or right.
Hourglass network:

The hourglass network has a personality of operating with the previous features generated by copying or adding by indices. The hourglass represented in the thesis paper gathers both the characteristic of a U-net and a Segnet NN. Copying the previous features to the matching up pooling process is a characteristic from the U-net architecture. Hourglass utilizes the previous feature as indices which originated from the Segnet network. Relu operates as the activation function in this hourglass network and most of Segnet and U-net architecture utilizes Relu too [13, 14]. Hourglass network delivers an output of key points on the input image. Each different key points are not distinguished into individual objects. Other algorithms for associating those key points must be included to determine the characteristic of the pig. Figure 8 shows the hourglass network structure.
Figure 8. Hourglass shape network. Hourglass network is based on the max pooling indices of Segnet and feature copy of the U-net.

PROBLEMS

The method for instance-level detection of multiple pigs in group-housed environments utilizes the Hourglass network which generates specific key point location of the targeted object. An algorithm must accord for calculating the connection of each key points generated. During the process of relating the points an error might occur of having the wrong object points connecting each other. From the paper of ‘Multi-Pig Part Detection and Association with a Fully-Convolutional Network’, 16 different channels which four of them representing the key points are generated as the location of the pig. The rest of the channels provides the characteristic of the connected key points [3]. Figure 9 shows the result of the targeted mapping after operating with the Hourglass model.
Figure 9. Four-channel output from the input image [3]. The input image is being processed through an Hourglass shaped network which generates the result of the target mapping as shown in figure b.

The structure of the four key points channels are represented as below.

\[ N = \text{number of pigs} \]

\[ n \in \{1 \ldots N\} \]

\[ t_n = (x_{tn}, y_{tn}), \text{representing the tail coordinates.} \]

\[ s_n = (x_{sn}, y_{sn}), \text{representing the shoulder coordinates.} \]

\[ l_n = (x_{ln}, y_{ln}), \text{representing the left ear coordinates.} \]

\[ r_n = (x_{rn}, y_{rn}), \text{representing the right ear coordinates.} \]

To approximate the level of uncertainty inherited by the user annotations of each body part location, parts within the target mapping were each represented by 2D
Gaussian kernels [3]. The 12 channels are utilized additionally in the target output to encode body part locations with 2D vector offsets to other body parts belonging to the same animal [3]. Total of 6 parts pairs that exist between the four parts and the target output only represents three in order to reduce unnecessary redundancy [3]. Figure 10 represents the 21 channels represented as three parts of associations. Figure 11 reveals the flow diagram converting the 16 channels image space representation to a set of 2D coordinates of each visible instance [3]. The table below shows the characteristic of the three pairs of vectors [3].

<table>
<thead>
<tr>
<th>Channels 1-4</th>
<th>Left Ear to Shoulder</th>
</tr>
</thead>
<tbody>
<tr>
<td>Channels 5-8</td>
<td>Right Ear to Shoulder</td>
</tr>
<tr>
<td>Channels 9-12</td>
<td>Shoulder to Tail</td>
</tr>
</tbody>
</table>

Table 1. Paired vector table [3].
Figure 10. Channel outputs.

Figure 10 (a) represents the input image. A 12 channels output from figure 10 (a) is generated as in figure 10 (b). The vectors joining three pairs of body parts are encoded into circular regions in channels 5 to 16 of the output [3] in figure 10 (b). Figure 10 (c) illustrate the locations of the vector encodings with their magnitude and directions [3]. The different colors being implemented depends on the magnitude and the directions [3]. Figure 10 (d) provides the mapping between the vectors and colors implemented [3].
Some problem occurs during the grouping process of the different key points. Figure 12 generates an example while the grouping is created by the Euclidean distance. Other algorithm for grouping purpose can be utilized but the lack of processing through a grouping mechanism will always deliver a certain possibility of an error. The vector approach with different channels provides a limit because key points found are not differentiated into individual objects.
Figure 12. Problem associating the nearest body parts utilizing Euclidian distance. As shown in the picture above, different key points of the pigs are associated together. The key points should have been associated with the three points of green, red, blue of the lower part of figure (b) with the yellow point of the upper part from figure (b).
NETWORK DESIGN

Our Approach extracts the data from the Mask R-CNN and convert it into matching dataset for training into the VGG-16 neural network. This idea came from removing the group attaching approach of different key points generated from the Hourglass network. In order to remove the grouping approach, we extracted the characteristic of the Mask R-CNN which generates an RPN classifying individual objects. It is important to reference that we utilize the characteristic of object differences for emphasizing other various NN having the possibility to implement different pose estimations. Our method generates a pose estimation of the pig with the VGG-16. An ROI and a binary mask data are generated from the Mask R-CNN and it is possible to extract those data individually. Figure 13 Shows the network flow process of the proposed model.

Figure 13. workflow process of the proposed approach. Extract the matching ROI, Masking data and the max height and width of ROI. Based on the matching input image and extracted data, generate the mask RGB image with data augmentation which later will be passed through the VGG network.
With the extracted information of Mask R-CNN, we generate a mask RGB image information with the matching ROI. The cutting result with the matching ROIs, input images and mask images produce the mask RGB images. We multiply the sliced ROI RGB image with the edited ROI mask image creating a mask RGB image. We extract the max height and width among the ROI and add zero padding to the mask RGB image with the obtained max height and width. Zero padding permits for the input of VGG NN to have consistency. VGG NN does not allow different input sizes.

For the binary mask data, a masking augmentation has been applied in order to gather better features around the surrounding object after generating the mask RGB image. The mask augmentation process is generated by comparing if the current pixel represents zero or one. A loop travels through the pixel of the mask data, and if the current pixel signifies one, we convert the upper and left pixel from the current location. The augmentation loop processed 20 times with the binary mask. Figure 14 delivers the procedure of masking augmentation.
Figure 14. Example of masking augmentation. The image shows a portion of binary mask pixel value. Based from current location of the pixel compare if the current value is one or zero. If the value is confirmed as one change the upper and left pixel value to one.

A blurring effect was implemented around the surroundings of the mask RGB images with the Gaussian filter. The surroundings of the mask RGB image is produced by subtracting the augmented mask (aug mask) and the original mask (org mask). We process the ROI RGB image with the Gaussian filter and multiply with the surrounding mask generated. The ROI RGB image multiplies with the org mask and the result produced adds with the multiplied surrounding mask. Figure 15 shows the process of making the augmented mask RGB image with Gaussian blur effects.
Figure 15. Development of augmented mask RGB image.

Based from figure 15, each image from each line from left to right gives a required result with the equation operated. Figure 15 (a) represents the augmented mask with the same method shown from figure 15. Figure 15 (b) represents the original mask with the ground truth data. Figure 15 (c) represents the result of surrounding mask with the subtraction of figure 15 (a) and figure 15 (b). Figure 15 (f) represents the Gaussian blur surrounding mask with the multiplication of figure 15 (d) of surrounding mask and figure 15 (e) of the Gaussian blur on ROI image. Figure 15 (g) signifies the original
image and Figure (h) represents the ROI RGB image. The multiplication result of figure 15 (g) and figure 15 (h) is shown as figure 15 (i) as mask RGB image. The result of mask RGB image and Gaussian blur surrounding image are added together as figure 15 (l).

After implementing the mask RGB image with Gaussian blur effects, data augmentation is applied to the data which creates various mask RGB datasets with different rotation. Data augmentation provides us to create diverse data and have a better efficiency of training network with VGG-16. Figure 16 generates an example of augmented datasets.

Figure 16. Example of augmented data.

Figure 16 (a) represents an example of an augmented mask generated with the gaussian blur surroundings. Figure 16 (b) signifies a horizontal flip of figure 16 (a).

Figure 16 (c) signifies a vertical flip from figure 16 (a). Figure 16 (d) shows a 90-degree
rotation of figure 16 (a). Figure 16 (e) represents a 90-degree rotation with a flip from figure 16 (a).

We utilized Keras Tensorflow as an open source tool which provides us libraries and functions for neural network purpose [21, 22]. Keras is a high level NN library written in python which works as a wrapper to Tensorflow [21]. Keras is developed based on user friendly and easiness of model building [21].
EXPERIMENT RESULT

The dataset was generated from [http://psrg.unl.edu/Projects/Details/12-Animal-Tracking](http://psrg.unl.edu/Projects/Details/12-Animal-Tracking) giving 2000 images of pigs based on top down view. 230 images out of 2000 were utilized as the dataset. The labeling ground truth information was generated utilizing the supervise.ly ([https://supervise.ly/](https://supervise.ly/)) software tool. Total of 230 images, each of them having 14 pig objects on average was labeled individually. 15,465 individual mask RGB image of pigs were created out of the 230 images. 4735 mask RGB images were categorized each as the pig standing up or laying down. It is important that the training dataset have the same number of proportions of classified data, if not the training will not generate a stabilized model which could deliver only one classification gathering the right result. As mentioned earlier the output of the operated VGG-16 network distinguishes as 2 classification with the sigmoid activation function. The output classification layer differentiates as [1, 0] for pig standing up and [0, 1] for laying down. Based on the sigmoid activation function, the output of the model will deliver an approximation value between 0 and 1 for each binary value. The higher the probability is closer to [0, 1] represents that the pig is laying down, [1, 0] representing the opposite. Each 4735 mask RGB images of pig standing up and laying down has been trained into the VGG-16 neural network providing a result of 76.87% accuracy. Figure 17 shows the results of the VGG-16 network trained with 9470 mask RGB images.
Figure 17. Model accuracy and loss of the augmented masked RGB image training. 100 epochs were run on the training with a batch size of 28. 70% of the 9740 masked RGB image which provides 6629 masked RGB images were set as training datasets, 2841 of the rest mask RGB image were set as validation datasets.

The 1270 of mask RGB image out of 15,465 images was provided as test datasets. The accuracy of the test dataset was generated as 74.13 %. 337 mask RGB images out of
the test datasets delivers false result from the trained model. Figure 18 offers some of the false result from the trained model.

<table>
<thead>
<tr>
<th>Input image</th>
<th>True label</th>
<th>Predicted label</th>
<th>Test data output</th>
</tr>
</thead>
<tbody>
<tr>
<td>Standing up</td>
<td>Laying down</td>
<td>[0.2, 0.78]</td>
<td></td>
</tr>
<tr>
<td>Standing up</td>
<td>Laying down</td>
<td>[0.25, 0.73]</td>
<td></td>
</tr>
<tr>
<td>Laying down</td>
<td>Standing up</td>
<td>[0.61, 0.42]</td>
<td></td>
</tr>
<tr>
<td>Laying down</td>
<td>Standing up</td>
<td>[0.51, 0.45]</td>
<td></td>
</tr>
</tbody>
</table>

Figure 18. Results of miss classified data based on the trained model. The model output produces a probability output between 0 and 1 for each binary value.

As the view is generated only from a top down view the characteristic of the pigs is limited to be determined. One of the features of determining if the pig is standing up or laying down depends on the location of the pig’s legs and how the pig’s legs are
represented. Based on the view, most of the common cases which the pigs are standing up is when the legs are covered from the body parts of the pigs. Another common case which the pigs are standing up is when the legs are shown to be straighten from the ground. The problem of differentiating the characteristic of the pigs occurs when the pigs which are laying down shows a similar characteristic of the pigs which are standing up.

A better approach would be having multiple information based on different angle views. The model generates some limitations because the features of the pig’s legs is difficult to differentiate based on a single point of view.
CONCLUSION

The main approach of this thesis applies the characteristic of Mask R-CNN which differentiate each individual object and present a classification method based on the pig’s behavior with the VGG-16 network. It is important to refer the RPN, the key algorithm distinguishing individual objects. Nonetheless there exists different algorithm such as You Only Look Once (Yolo) for object detections [15]. Yolo delivers the object detection based on regression problems which divides the image into an N by N grid and for each grid cells providing prediction bounding boxes [15]. Mask R-CNN offers a similar approach with the Hourglass network which creates K number of key points of an instance [4]. The K key points are still treated independently [4]. Pose estimation always follows a characteristic of key point prediction based on the neural network. This characteristic could be discovered with hand pose estimation [16] or human pose estimation using Deep neural networks [17].

Overall, the result of this thesis generates the possibility of differentiating various of pigs depending on their characteristic. Instead of implementing other algorithms for associating the several key points and seeking their features, it is possible to distinguish individual pigs and classify their characteristics.

The provided approach has some limitations based on the datasets. The extracted masked RGB image was only generated from a top down view and the characteristic of the pig object can sometimes be hard to distinguish based on this approach. Adding different angle views of datasets could offer better characteristic of the pig. The combination within the mask RGB images and key points joints could deliver better pose
estimation too. This method is not limited for the pig pose estimation but for other types of classification if necessary.
REFERENCES

   Automated tracking to measure behavioural changes in pigs for health and welfare
   monitoring. *Scientific Reports, 7*(1). doi: 10.1038/s41598-017-17451-6

   Automatic Individual Pig Detection and Tracking in Pig Farms. *Sensors, 19*(5),
   1188. doi: 10.3390/s19051188

   Multi-Pig Part Detection and Association with a Fully-Convolutional

4. Kaiming He, Georgia Gkioxari, Piotr Dollar, Ross Girshick. (2017). Mask R-
   CNN. 2017 *IEEE International Conference on Computer Vision (ICCV).* doi: 10.1109/iccv.2017.322

5. Shaoqing Ren, Kaiming He, Ross Girshick, Jian Sun. (2017). Faster R-CNN:
   Towards Real-Time Object Detection with Region Proposal Networks. *IEEE
   Transactions on Pattern Analysis and Machine Intelligence, 39*(6), 1137–1149. doi:
   10.1109/tpami.2016.2577031

6. Tsung-Yi Lin, Piotr Dollar, Ross Girshick, Kaiming He, Bharath Hariharan, Serge
   Conference on Computer Vision and Pattern Recognition (CVPR).* doi:
   10.1109/cvpr.2017.106


