Cat Recognition Based on Deep Learning

Pengyin Chen, Arial Xiao Qin, Junhao Lu

Cat Recognition Based on Deep Learning

Pengyin Chen
Department of Computing Science
University of Alberta
Edmonton, Canada
pengyin@ualberta.ca

Arial Xiao Qin
Department of Computing Science
University of Alberta
Edmonton, Canada
xqin5@ualberta.ca

Junhao Lu
Department of Computing Science
University of Alberta
Edmonton, Canada
junhao3@ualberta.ca

***Abstract*—Animal recognition and identification research is critical for animal protection and monitoring. We focused specifically on cats in this report. A method for cat recognition and identification is proposed that is based on Autoencoders and Convolutional Neural Networks (CNN). Additionally, we created a fresh dataset containing 1,994 photos of 17 cats. This article details how to utilize Autoencoder to denoise cat image data and combine it with CNN to create a strong model for the same cat recognition, which serves as motivation and a reference for future research in animal recognition.**

***Index Terms*—Animal Management, Cat Identification and Recognition, Autoencoder, CNN**

I. Introduction

As computer vision technology advances and mobile camera devices become more widespread, this trend has resulted in an increase in the number of computer vision apps for object detection.

Animal detection and recognition are becoming increasingly important in a variety of industries, including animal husbandry, pet care& lost lost and found, as well as wildlife conservation. For instance, recognition in animal husbandry is dependent on the printed tag connected to the animals. Real-time animal recognition in farms, on the other hand, is dependent on the tracking of human eyes. When it comes to pets, owners typically implant intrusive microchips to deter theft or loss. The procedures outlined above may be costly and time-consuming. With facial recognition for cows, it is possible to monitor the amount of water and food consumed by the cow throughout the day, as well as detect signs of illness and aberrant behaviour [1].

The advancement of computer vision and the availability of affordable photographic equipment enables the automatic recognition of objects. [2] For feature detection and extraction, animal recognition makes use of feature descriptors. These attributes are used to train prediction models that can ultimately predict the label for a given image. We will discuss current advancements in image preprocessing, feature extraction, and classification in this paper. Additionally, this report will demonstrate the advancement of object recognition by deep learning.

Identify applicable funding agency here. If none, delete this.

II. Literature review

A. Image Preprocessing

Image preprocessing aims to enhance features and decrease the impact of distortions for further image processing. As for animal recognition, the images usually have the characteristics of uneven illumination, complex background and image noises [3].

Noises are present in the digital images, which can decrease object recognition accuracy without appropriate noise reduction. Gaussian noise frequently happens in animal images because the animal images might be taken in poor illumination conditions such as farms, fields, and shelters. The gaussian filter was developed for Gaussian noise reduction by performing convolution between a convolution kernel and image [4]. Recently, a low-pass filter called Gaussian pyramid was applied to cattle recognition by removing noise at different Gaussian pyramid levels [5].

In images, the background portion might be much greater than the target of interest, which could eventually mislead the model to learn background features. Therefore, cropping out the unnecessary background could improve further image recognition accuracy. In addition, as for the CNN model, cropping the image into the same size is essential for model training. Until now, this operation has been proved to improve the accuracy of sheep and pigs recognitions [6], [7]

B. Feature Descriptors

The previous image recognition works involved three feature extractions, including low, mid, and high-level methods. Low-level feature extraction mainly focuses on the local feature by evaluation of the small pieces of an image (e.g. a group of pixels). Currently, scale-invariant feature transform (SIFT) descriptor and histogram of gradient (HOG) descriptor are frequently used for object recognition as low-level approaches. The mid-level approach aims to use statistical analysis of local features to represent a global image. [8] For example, the bag of words (BoG) algorithm is one of the most used approaches in image classification. High-level methods usually utilize deep learning methods to describe the entire image. [9]

SIFT descriptor utilizes gradient distribution in the target regions, which is invariant to the change of rotation, location and scale to find the unique feature points. The SIFT

algorithm comprises four major steps: feature point detection and localization, followed by orientation assignment and feature descriptor generation. To improve feature description efficiency, gradient location and orientation histogram (GLOH) was proposed as a SIFT-like descriptor. GLOH descriptor decreases the size of grids by principal component analysis (PCA) and adjusts the grid location to enhance distinctiveness and robustness. However, the GLOH descriptor has not been applied in animal recognition, while SIFT descriptor is frequently used (e.g. cattle, cow). The HOG descriptor relies on the histogram valuation of gradient orientation occurrences in the grid of a given image [10]. The shapes and edges of images can be descriptive by evaluating the local gradient. HOG approach was frequently applied for object recognition because it can represent the contour of objective and invariant to the change of illumination [11]. Meghna and his team proposed a novel method for the recognition of human pose using the Radon transform. They create a way to present the human skeleton as a binary vector and utilize parametric Radon transform to extract pose features, which generates maximum corresponding to specific orientations of the skeletal representation. [12] Also, in order to get the feature, we still have certain limitations like shape of the objects or image noise. Their team represented Gaussian and Laplacian of Gaussian weighting functions for robust features to overcome and suppress the noise and create a new mechanism for determining the optimal weighting function based on image parameters, more specifically the edge characteristics of objects in the image. [13] Similarly, Mark and his team also proposed a new method to solve the feature detection in non-SVP(single viewpoint) situations. They represent a new mathematical model to detect features in panoramic non-SVP images using a modified Hough transform and significantly improve the performance in identifying line features with only estimated calibration. [14]

For face recognition, one of the most important aspects is finding the landmark points and presenting the image as a vector to feed the neural network for training purposes. Although we are researching cat faces, some work has been done on human faces that could also give us some inspiration on how to handle cat faces more efficiently. Lijun and his team created an MPEG4 face modelling using fiducial points. They utilize two views of a person's face and the predefined fiducial points to generate a 3D face model. [15] Also, feature descriptors could be more accurate if we apply them into some specific significant part of the face. Lijun and his team have built a shape-adaptive model to track noses based on the nose shape estimation. A two-stage region growing method was applied to certain areas, and used pre-defined templates to extract the shapes of the nostril and nose-side. Finally, the extracted feature shapes are exploited to guide a facial model to complete an accurate adaptation. [16] In order to make the model match the face more accurately, Lijun and his team used a partial texture updating method for realistic facial expression synthesis with facial wrinkles. First, they use a color-based deformable template matching method to estimate fiducial points on a face. Second, an extended dynamic mesh matching algorithm is developed for face tracking. Next, textures of interest (TOI) in the potential expressive wrinkles and mouth–eye texture areas are captured by the detected fiducial points. Among the TOI, the so-called active textures or expressive textures are extracted by exploring temporal correlation information. Finally, the entire facial texture is synthesized using the active texture. [17]

Overall, based on the literature reviews for different methods, there is a brief summary for some of the traditional feature extraction methods, and Convolutional neural network (CNN). The traditional feature detection methods are Harris Corner Detection, SIFT, Local Binary Pattern (LBP), speeded Up Robust Feature (SURF) and a few more, lately it has been replaced by CNN, since CNN is automated with high accuracy to extract complex features with higher efficiency.

SIFT is one of the oldest methods started in 1997. It is very reliable to get scale-invariant results by looking for features on the entire space. It has a good accuracy and the result features are robust, while the time complexity is comparely slow and computational heavy. Also, it has poor performance when extracting features from smoothing targets. LBP was introduced in 1994. It is a simple but effective image operator that labels pixels of an image and compares its neighbors. LBP has a good performance under translation and grey scale. Although the LBP method is computational light, it is not invariant to any image rotation and scales. [18]

SURF is similar to SIFT but much faster, while it has a lower accuracy compared to SIFT. The main drawbacks for SURF, compared to SIFT, is the same, that it cannot detect objects accurately on textureless objects. [19]

CNN is a deep learning neural network and could take a pre labeled image dataset as input, train the model with the dataset and be able to classify new images. It has automatic feature extraction functionality and has the highest accuracy, while it requires a large dataset for training and is computationally intensive. There's a few common convolutional networks used for image classification, one is ResNet and another is VGG architecture. [20]

Iris recognition is another automated method that has been widely used for biometric identification. It specialize in pattern recognition techniques in images, and find similarity in the 2 images that is extremely difficult for other recognition method to achieve the same, such as human eyes that has unique and complex pattern, yet stable and can be visually seen. [21]

On top of the above methods, SVM (Support vector machines) is often being used together to further improve the accuracy of the traditional methods. The conducted research in Iris Image Segmentation showed that it achieves highest accuracy with SVM on traditional methods (99.92%). [22], [23]

C. Learning Methods

The classical approaches based on the hand-craft feature descriptors and classifiers have been making remarkable achievements in image recognition. However, image descriptors usu-

ally need to be engineered or designed manually. It takes a long time to evaluate the accuracy of descriptors. The CNN method was started in 1980. Due to the computational power limit at the time, it was not popular as it required a large amount of data for training. Meanwhile, classifier-based methods including K-means and SVM had better performances than CNN approaches. These problems were solved by the deep brief network algorithm and utilization of GPU. CNN based image recognition now dominates the image classification with high accuracy [Semantic Learning for Image Compression]. The research paper for wrinkle detection, which builds image classification models based on convolutional neural networks, has achieved 85.9% accuracy for wrinkle detection. [24]

GoogLeNet is a deep neural network model based on the Inception module launched by Google. It won the championship in the ImageNet competition in 2014. [25] It has been improving in the following two years, forming Inception V2, Inception V3, Inception V4 and other versions. In order to solve the issue caused by 1) too many parameters 2) computational heavy caused by too many network layers 3) Gradient dispersion problem, the notion of Inception was introduced. Inception is to put multiple convolutions or pooling operations together to form a network module. When designing a neural network, use the module as a unit to assemble the entire network structure. Before Inception was introduced, neural networks would use the same size convolution layers. However, images with different scales need different sizes of convolution layers and could have different performances under different sizes. An Inception module provides multiple convolution kernel operations, and the network could choose to use it by adjusting the parameters during the training process. GoogleNet has a wide range of applications in animal detection and recognition. In the competition of ImageNet, it improved the accuracy from 22.6% to 43.9%, and two years later, ImageNet stopped its future competition, which means GoogleNet has reached the milestone that machines could have the accuracy as close as humans in the area of classifying images. AlexNet was proposed by Alex Krizhevsky in 2012 and won the 2012 ILSVRC competition. [26] It uses ReLU as the activation function of CNN instead of using Sigmoid and successfully solves the issue of Gradient dispersion caused by Sigmoid. During training, it uses Dropout to ignore some neurons to avoid model overfitting randomly. Before AlexNet, most CNNs used average pooling. However, AlexNet uses overlapped max pooling to avoid the blur effect of average pooling and enhance the richness of features. It proposed a new LRN (Local Response Normalization) layer to create a competition system for local neurons to suppress partial inactive neurons and improve the model's generalization ability. Also, data enhancement adds over 2000 times the amount of the original data to avoid overfitting. AlexNet could be applied to animal facial recognition. Using the structural similarity between animal faces and human faces, we could build a mapping network from animal faces to human faces (the first five convolution modules of AlexNet) and use facial landmark points to localize and recognize animal faces.

III. Method

The execution of experiments were performed with Core-i7-9700 CPU and GTX 2070 GPU, based on Python-3.7, Tensorflow gpu-2.1.0, cuda-10.1.243, cudnn-7.6.5, and function library Opencv python-3.4.2, Numpy-1.18.5, Matplotlib-3.2.2 and Scikit learn-0.23.1

A. Dataset Preparation

We started with photos from Kaggle's cat dataset [27]. This dataset comprises almost 20,000 photos of cats with varying image quality. The majority of cats in the collection are unique; only a few cats have two or three photos with varying attitudes and backgrounds, as illustrated in figure 1. Despite the fact that the Kaggle cat dataset has a significant number of cat images, we quickly concluded that the Kaggle cat dataset did not fit our purpose. Our objective is to identify the same cat in a variety of stances and environments. Therefore, a new dataset that contains different cats, and each cat has multiple images, is required for continuing the project.

Fig. 1. Example of cat images in Kaggle cat dataset [27]

1) Data Augmentation: We employ the data augmentation technique to enrich existing photos in order to satisfy dataset requirement for the project. We randomly choose ten different cats with high-quality images. Following that, we call OpenCV library methods to execute grayscale, mirror, and angle rotations. Each cat image generates more than ten other images. The augumented images are used to create a fresh dataset for training and testing CNN models.

2) Build Dataset: This dataset's purpose is to collect enough photos of the same cat. Images are gathered through video and Instagram cat bloggers. The script of Instaloader downloads images. Furthermore, we choose and download cat videos from https://www.pexels.com. Then, using the OpenCV library's video capture function, we randomly selected and captured video frames.

3) Cat Face Detection and Crop: We intend to clip out the cat face from the entire image in order to reduce the influence of the background and the computational load on the CNN model. The Haar-Cascade and HOG detector algorithms are used to achieve cat head identification and background removal. The HOG recognition algorithm is created from

scratch, whereas the Haar-Cascade approach is adapted via the OpenCV library's function. Finally, we purge the dataset by manually removing photos that do not contain cat heads.

4) Dataset Finalization: We enhance the Kaggle dataset using OpenCV library methods to create several photos of the same cat, as seen in Figure 2. However, the CNN model is incapable of encoding an object's position or orientation. As a result, this approach is not used to generate the dataset.

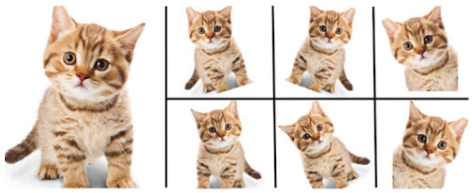

Fig. 2. Examples of data augmentation

Using OpenCV video capture and Instagram image download. We create a dataset with a total of 10,000 photographs. Some photographs in this dataset contain a significant portion of background or other undesirable objectives. To avoid these potential influences on model training, we use Haar-like feature classifiers and HOG to clip the cat face out of the entire image. In comparison to the HOG approach, the Haar method was more prone to misclassifying other targets (e.g., grass) as cat faces, as shown in Figures 3A and 3B. As a result, we use HOG to go through the entire dataset and manually delete the incorrect photos. Finally, our dataset has 17 different cats with 1994 photos with a resolution of 150*150. However, some photographs contain other types of noise, such as little dots, occlusion, or grass. Following that, the dataset was divided into three parts: training, validation, and testing, with a split ratio of 7: 1: 2. In addition, the appropriate label for each image data was included to assess the accuracy.

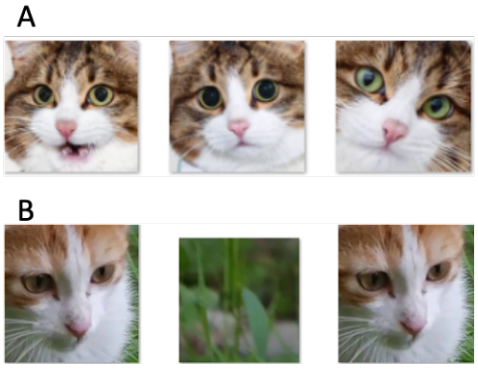

Fig. 3. Image operated by HOG (A) and Haar-like feature classifier (B)

B. CNN Model

1) Keras-based CNN model: Our dataset contains all different types of cats, and the majority of them have extremely similar appearances and have a common trait in their body parts, which might make categorization much more difficult. CNN's feature map (convolution layer) may be able to help us by determining the most correct feature dimension for classification.

Using the Keras framework, we construct our CNN model and define appropriate parameters from scratch. The CNN model begins with a rectifier implemented with the Relu function to facilitate gradient propagation. Following that, a convolution layer with the appropriate filter numbers and block size was added, as well as an activation function and a max-pooling layer. It's also worth noting that we include the dropout layer after the max-pooling layer to increase the robustness of our model, as recommended in the original dropout study. [28]. The architecture of our CNN model is shown in Figure 4.

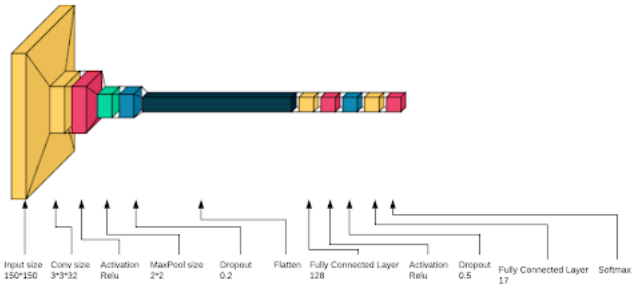

Fig. 4. Visualization of CNN architecture

Optimizers are responsible for minimising the cost function in CNN models. We investigate three Keras-implemented optimizers in this paper. The Adadelta optimizer employs a stochastic gradient descent algorithm. This optimizer is capable of resolving two issues: continuous decay of the learning rate and manual setting of the global learning rate. Adam optimizer makes use of stochastic gradient descent and moment estimation at first- and second-order. [29]. SGD optimizer in the Keras framework construct a stochastic gradient descent optimizer with momentum. The momentum function can assist a model in moving in the desired direction and also attenuate oscillations. [30]

2) Autoencoder Implementation: We updated our model and added a new Autoencoder Module to handle the issue of image noise. We use the Autoencoder for the train dataset to denoise the images. We begin by training the Autoencoder model on a subset of each dataset (around 10%). Then, we add noise to the training data and feed it into the encoder-decoder model to train our Autoencoder model. Then, once the Autoencoder model has been trained, we load all the data into it to generate denoised images. Finally we analyse the influence of image noise on our CNN model using the denoised images generated from last step.

IV. Result

A. CNN Model

After constructing the dataset and CNN model, we begin testing the CNN model. We begin by optimizing with Adadelta

and setting the learning rate to 0.001 for model testing. As illustrated in Figure 5, after 400 epochs in the testing set, this CNN model is unable to converge. Additionally, the accuracy cannot be increased above 80%.

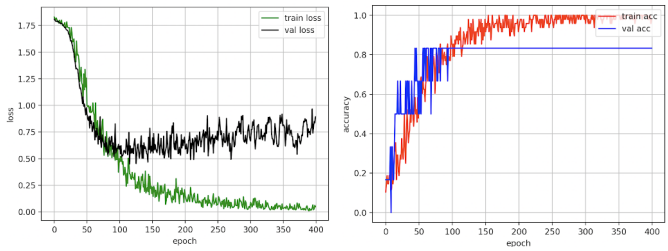

Fig. 5. Preliminary results of CNN model (learning rate = 0.001, Optimizer = Adadelta) Left image: Curve of Loss Value Right image: Curve of Accuracy Value

Additionally, we discovered that our CNN model can converge based on this preliminary test. As a result, it is unnecessary to add more layers to the model. As a result, we begin by adjusting the learning rate and comparing various optimizers in order to improve the model's performance. We discovered that setting the learning rate at 0.01 is reasonable after multiple attempts and experiments.

As shown in Figure 6, the convergence rate of the Adadelta based model becomes much higher than before. And the final result of Adadelta optimzer is good. However, the SGD based model was very fast as well as the improvement of accuracy. It might result in overfitting problems. Compared with SGD and Adadelta, Adam optimizer's performance is not stable, especially when training for many loops, it's accuracy various in a quiet wide range, even though the average of the accruracy does not change much. Therefore, it is appropriate to choose the Adadelta as optimizer in our study.

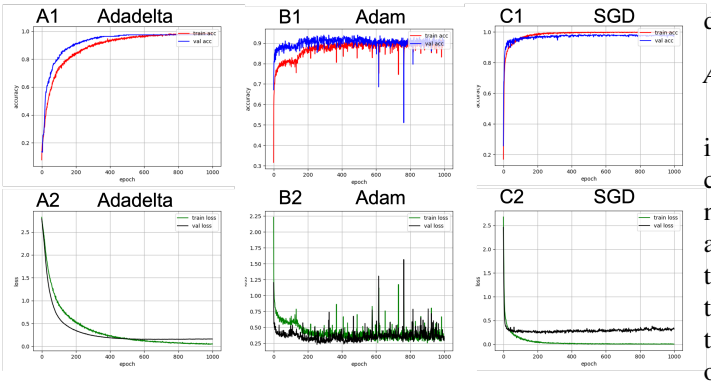

Fig. 6. Loss value accuracy value curve graphs of different optimizers (learning rate = 0.01) (A1) Accuracy value curve of Adadelta optimizer (A2) Loss value curve of Adadelta optimizer (B1) Accuracy value curve of Adam optimizer (B2) Loss value curve of Adam optimizer (C1) Accuracy value curve of SGD optimizer (C2) Loss value curve of SGD optimizer

Additionally, we examine how the Autoencoder module's implementation affects the accuracy of our CNN model. In our test, the average time required to provide classification results for the training dataset is 421 seconds. After 7,000 epochs of training processing, we have a decent outcome. The error rate is approximately 0.03 after 1,000 epochs. The error rate is approximately 0.02 after 3,000 epochs. After 7,000 epochs, the error rate approaches 0.01. As our final evaluation revealed, it does give a reasonably good performance, as illustrated in Figure 7.

Fig. 7. Sample images from Autoencoder Module output

In comparison to the single CNN technique, we incorporate the Autoencoder module on top of single CNN technique. Comparing the test result without Autoencoder, with the same dataset, we discover that our technique performs far better when dealing with noisy photos. Without Autoencoder, many cat faces are undetectable because of the image noise, such as occlusion or minor irrelevant objects. And false positive prediction labelling could occurs as well. We can easily reduce noise with Autoencoder and ensure that our prediction labels fall into the correct cat category as much as possbile. While no model is flawless and there are still some defects in the result prediction, we can see that there is more accurately predicted labelling while training the same number of epochs as before, indicating that our model has become more robust.

V. Discussion

In this project, we faced challenges in dataset preparation and CNN model optimization. Therefore, we would like to discuss our rationale for solutions.

A. Rational of dataset preparation

To address the problem of recognising the same cat, it is important to collect sufficient photographs of the same cat in various angles and poses. However, there is currently no open-source dataset that meets this requirement. Data augmentation is a frequently used technique for reducing the time required for data collecting. For image modification, the OpenCV library includes numerous operations such as translation, rotation, and reflection. The CNN model, on the other hand, is unable to decode location information since the convolutional layers can only detect low-level features (e.g. edge gradients). Additionally, it has been noted that Haar-cascade detection is prone to producing false-positive recognition results. For example, the Haar-cascade detector is easy to recognize unwanted objectives when dealing with images that do not have front cat faces. Additionally, the Haar-cascade is ineffective when confronted with occlusion issues. By contrast, because the HOG method is based on the identification of the contour of target objects, it may recognise

non-front faces and faces with occlusion. [31]. Based on our attempts and experiments with the Autoencoder module, we find out that Autoencoder could provide our model a certain level of denoise ability and during the training process, we can see that validation accuracy is higher than the training accuracy. However, we also notice that it eliminates a portion of the feature information from image data, which leads to low accuracy in later CNN evaluation.

B. Rational of CNN model construction and optimization

In order to solve the problem of recognition and identification of cats, we need to treat this problem as a classification problem. The method we first found is CNN, which has a wide range of applications in image classification. This model has been used in competitions like ImageNet and has shown us its performance in detecting and classifying images. CNN normally has a better performance in classification of different categories of images like dogs and cats. However, our dataset contains all kinds of cats and most of them have very similar outlooks and share the common properties in their body parts, which could make the classification much more difficult. CNN's feature map (convolution layer) could provide us with a solution by finding the most accurate feature dimension for classification.

In our initial test, we choose the Adadelta optimizer and set the learning rate as 0.001. However, the curve of loss graph demonstrates that the model was at the local optimal. Although increasing the learning rate to 0.01 could solve this problem, the convergence rate of the Adadelta optimizer is still low. SGD optimizer changes and updates the gradients for each training dataset. In addition, the momentum function is integrated into the SGD optimizer in the Keras framework. The momentum function can help the gradient move in the right direction.

VI. Conclusion and Future Work

In conclusion, we build up our own image dataset and construct the CNN model from scratch based on the Keras framework. In addition, we explore the function of the Autoencoder in our model and compared different optimizers.

For future work, we can expand our dataset with more cats that have more similar cat faces and body colors so that it could simulate the situation in real life, which requires our model to extract the most distinguishable feature map during the training process. In addition, it is necessary to explore how many images are required for model training and optimize the dataset split ratio. Because it might be hard to collect adequate images as this dataset for model training.

Also, in order to identify a new cat, the current model will need to add the new cat to the dataset and train it again. Instead, we could incorporate the idea of incremental learning so that a new cat category could easily and fastly be added to our model. The Autoencoder could provide our model the ability to process image data with noise, but we need to improve our Autoencoder module so that it will eliminate the noise and could keep the original image features after denoise.

VII. Roles

We wrote the proposal, literature review, and report together.

A. Dataset preparation

- **Data augmentation**: Arial Qin & Pengyin Chen
- **Image collection and cleanup**: Arial Qin & Junhao Lu
- **Face detection**: Pengyin Chen & Arial Qin
- **Autoencoder implementation**: Pengyin Chen & Junhao Lu

B. CNN model construction and optimization

- **CNN model build up**: Pengyin Chen & Junhao Lu
- **Fine-tune of CNN model**: Junhao Lu & Arial Qin
- **Optimizer Comparison**: Junhao Lu & Arial Qin

VIII. Acknowledgment

We thank Dr. Anap Basu and Guanfang Dong's suggestion on dataset preparation and overall guidance throughout the project.

References

[1] Santosh Kumar and Sanjay Kumar Singh. Cattle recognition: A new frontier in visual animal biometrics research. *Proceedings of the National Academy of Sciences, India Section A: Physical Sciences*, 90(4):689, 2020.

[2] R. Szeliski. *Computer Vision: Algorithms and Applications*. Texts in Computer Science. Springer London, 2010.

[3] Fernandes Arthur Francisco Araújo, Dórea João Ricardo Rebouças, and Rosa Guilherme Jordão de Magalhães. Image analysis and computer vision applications in animal sciences: An overview. *Frontiers in Veterinary Science*, 7, 2020.

[4] Weizheng Shen, Hengqi Hu, Baisheng Dai, Xiaoli Wei, Jian Sun, Li Jiang, and Yukun Sun. Individual identification of dairy cows based on convolutional neural networks. *Multimedia Tools and Applications: An International Journal*, 79(21-22):14711, 2020.

[5] S. Kumar, A. Pandey, K. Sai Ram Satwik, S. Kumar, S.K. Singh, A.K. Singh, and A. Mohan. Deep learning framework for recognition of cattle using muzzle point image pattern. *Measurement: Journal of the International Measurement Confederation*, 116:1–17, 2018.

[6] S. Abu Jwade, A. Mian, and Guzzomi). On farm automatic sheep breed classification using deep learning. *Computers and Electronics in Agriculture*, 167, 2019.

[7] C. Chen, W. Zhu, M.L.V. Larsen, T. Norton, M. Oczak, K. Maschat, and J. Baumgartner. A computer vision approach for recognition of the engagement of pigs with different enrichment objects. *Computers and Electronics in Agriculture*, 175, 2020.

[8] Leng Chengcai, Zhang Hai, Li Bo, Cai Guorong, Pei Zhao, and He Li. Local feature descriptor for image matching: A survey. *IEEE Access*, 7:6424 – 6434, 2019.

[9] Bovw model for animal recognition: an evaluation on sift feature strategies. 2015.

[10] Krystian MIKOLAJCZYK and Cordelia SCHMID. A performance evaluation of local descriptors. *IEEE transactions on pattern analysis and machine intelligence*, 27(10):1615 – 1630, 2005.

[11] M.I. Patel, V.K. Thakar, and S.K. Shah. Image registration of satellite images with varying illumination level using hog descriptor based surf. In *Procedia Computer Science*, volume 93, pages 382–388, (1)Sankalchand Patel College of Engineering, 2016.

[12] M. Singh, M. Mandai, and A. Basu. Pose recognition using the radon transform. In *Midwest Symposium on Circuits and Systems*, volume 2005, pages 1091–1094, (1)Department of Electrical and Computer Engineering, University of Alberta, 2005.

[13] M. Singh, M.K. Mandal, and A. Basu. Gaussian and laplacian of gaussian weighting functions for robust feature based tracking. *Pattern Recognition Letters*, 26(13):1995–2005, 2005.

[14] M. Fiala and A. Basu. Hough transform for feature detection in panoramic images. *Pattern Recognition Letters*, 23(14):1863–1874, 2002.

[15] Lijun Yin and Anup Basu. Mpeg4 face modeling using fiducial points. In *IEEE International Conference on Image Processing*, volume 1, pages 109–112, Univ of Alberta, 1997.

[16] L. Yin and A. Basu. Nose shape estimation and tracking for model-based coding. In *ICASSP, IEEE International Conference on Acoustics, Speech and Signal Processing - Proceedings*, volume 3, pages 1477–1480, Department of Computing Science, University of Alberta, 2001.

[17] L. Yin and A. Basu. Generating realistic facial expressions with wrinkles for model-based coding. *Computer Vision and Image Understanding*, 84(2):201–240, 2001.

[18] Chengsheng Yuan and Q. M. Jonathan Wu. Fingerprint liveness detection based on multi-modal fine-grained feature fusion. In Troy McDaniel, Stefano Berretti, Igor D. D. Curcio, and Anup Basu, editors, *Smart Multimedia*, pages 417–428, Cham, 2020.

[19] Gabriel Lugo, Nasim Hajari, Ashley Reddy, and Irene Cheng. Textureless object recognition using an rgb-d sensor. In Troy McDaniel, Stefano Berretti, Igor D. D. Curcio, and Anup Basu, editors, *Smart Multimedia*, pages 13–27, Cham, 2020.

[20] Kushal Mahalingaiah, Harsh Sharma, Priyanka Kaplish, and Irene Cheng. Semantic learning for image compression (slic). In Troy McDaniel, Stefano Berretti, Igor D. D. Curcio, and Anup Basu, editors, *Smart Multimedia*, pages 57–66, Cham, 2020.

[21] Xianting Ke, Lingling An, Qingqi Pei, and Xuyu Wang. *Race Classification Based Iris Image Segmentation*, pages 383–393. 07 2020.

[22] Munawar Hayat, Stefano Berretti, and Naoufel Werghi. *Fused Geometry Augmented Images for Analyzing Textured Mesh*, pages 3–12. 07 2020.

[23] Xianting Ke, Lingling An, Qingqi Pei, and Xuyu Wang. *Race Classification Based Iris Image Segmentation*, pages 383–393. 07 2020.

[24] Abtin Djavadifar, Brandon Graham-Knight, Kashish Gupta, Marian Körber, Patricia Lasserre, and Homayoun Najjaran. *Robot-Assisted Composite Manufacturing Based on Machine Learning Applied to Multi-view Computer Vision*, pages 199–211. 07 2020.

[25] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going deeper with convolutions. 2014.

[26] A. Krizhevsky, G.E. Hinton, and I. Sutskever. Imagenet classification with deep convolutional neural networks. *Communications of the ACM*, 60(6):84–90, 2017.

[27] Chris Crawford. Cat dataset, Feb 2018.

[28] Geoffrey E. Hinton, Nitish Srivastava, Alex Krizhevsky, Ilya Sutskever, and Ruslan R. Salakhutdinov. Improving neural networks by preventing co-adaptation of feature detectors. 2012.

[29] E.M. (1) Dogo, O.J. (1) Afolabi, N.I. (1) Nwulu, C.O. (2) Aigbavboa, and B. (3) Twala. A comparative analysis of gradient descent-based optimization algorithms on convolutional neural networks. In *Proceedings of the International Conference on Computational Techniques, Electronics and Mechanical Systems, CTEMS 2018*, number Proceedings of the International Conference on Computational Techniques, Electronics and Mechanical Systems, CTEMS 2018, pages 92–99, (1)Department of Electrical and Electronics Engineering Science, University of Johannesburg, 2018.

[30] I. (1) Kandel, M. (1) Castelli, and A. (2) Popovič. Comparative study of first order optimizers for image classification using convolutional neural networks on histopathology images. *Journal of Imaging*, 6(9), 2020.

[31] D. (1) Zhou, J. (1) Wang, and S. (2) Wang. Countour based hog deer detection in thermal images for traffic safety. In *Proceedings of the 2012 International Conference on Image Processing, Computer Vision, and Pattern Recognition, IPCV 2012*, volume 2, pages 969–974, (1)Dept. of Mech. and Ind. Eng., University of Minnesota, 2012.