Journal of
Data Acquisition and Processing

ISSN: 1004-9037
https://sjcjycl.cn/
DOI: 10.5281/zenodo.7765926

ENHANCING FARM MANAGEMENT THROUGH DEEP NEURAL NETWORK FOR ANIMAL DETECTION

Mrs Indumathi R
Assistant Professor

Dr. N.Palanivel
Associate Professor

Sneha M
Rajalakshmi T.G

Shanmugalakshmi R
UG Scholars
Department of CSE, Manakula Vinayagar Institute of Technology, Puducherry, India

Abstract -- The aim of the research study is to address the numerous challenges associated with developing a productive approach for monitoring, categorizing, and identifying animals. Many computational models that can precisely recognize and track animals in various surroundings must be developed in order to create such a system.To this end, numerous techniques have been proposed and evaluated to improve the efficiency and accuracy of animal segmentation, detection, classification, and tracking.In particular, the proposed techniques have focused on separating animals from their surroundings, which is a critical step in the accurate categorization of animals.Two approaches were suggested to complete this job, and the effectiveness of the suggested animal segmentation algorithm was assessed using a region-based performance metric. The proposed classification model utilized a variety of features and classifiers to accurately categorize animals. Using segmented animal pictures, Gabor, color, and LBP were extracted, and the potential for combining the features to enhance classification performance was investigated.Furthermore, extracted features were represented in the form of interval-valued type data to preserve inter and intra-class variations of animals. The classification of animals was accomplished using symbolic classifiers and SVM, which allowed for the accurate categorization of animals based on their features. Finally, a model was proposed to segment, track, and label animals in videos, which utilized a region merging-based segmentation algorithm and the nearest neighbor classifier. This model was divided into three stages: segmentation, tracking, and labeling. In the first stage, an animal video was given as input, and frames were extracted from the videos. Segmentation was performed for all edges using a region merging-based segmentation algorithm. In the second stage, video frames were used to track the animal in entire animal videos, and Gabor features were extracted from the images and stored in the knowledgebase for labeling. The classification of animals was accomplished using the nearest neighbor classifier.Overall, the techniques and models proposed in this research have the potential to significantly improve the accuracy and efficiency of animal detection, classification, and tracking systems, thereby facilitating research in numerous fields, such as wildlife conservation, animal behavior research, and more.

Keywords -- Animal recognition system, LBPH, PCA, SVM, and neural networks.

I. INTRODUCTION

Over the years, the observation of various animals with different structures and forms has emphasized the need for classification, especially since over a million species have been identified. Animals can be classified based on various criteria such as their structure, DNA, color, shape, etc. However, the vast taxonomy of animals requires a guidebook for identifying an animal from images. Despite advances in digital and mobile technology, it remains difficult to identify animals based on images alone, which can be frustrating. The increasing usage of surveillance cameras in animal behavior studies necessitates the identification and tracking of animals in videos, which is a time-consuming and tedious manual process. An automatic system is needed to classify images or frames of animals based on their visual content. In this thesis, we propose and evaluate a system that can classify animals using novel algorithms for segmenting, tracking, detecting, and classifying different types of animals. We also design suitable features and classifiers to classify animals into respective classes and a tracking system to aid in the study of animal behavior. To test our system's effectiveness, we create a large data set of animals that includes both images and videos since there is no standard large data set of animals available in the literature. We present the results of our experimentation on this data set, demonstrating the effectiveness of our system compared to other existing models.

II. LITERATURE SURVEY

Xie et al. [1] conducted a study to investigate the use of dense feature extraction for improving the accuracy and robustness of instant recognition systems. They examined various methods to enhance precision and recall in object recognition tasks and proposed an ensemble-based approach that combines scores from multiple modalities. Through their experiments, the authors achieved a significant improvement over previously published results on the Challenge on Willow datasets. They attributed the success of their approach to the use of dense feature extraction and the implementation of multimodal feature models.Through the implementation of these techniques, Xie et al. were able to attain greater accuracy and resilience in their real-time recognition system. These results have practical implications for the advancement of more efficient and reliable algorithms for object recognition.

Eitel et al. [2] proposed a new architecture that combines RGB and depth information to recognize objects. Their architecture consists of two separate CNN processing streams that can handle depth data effectively. They also introduced a multi-stage training methodology and two key components that are essential for dealing with depth data in CNNs. Their approach was tested in challenging real-world settings, and they achieved promising results in object recognition tasks. In addition, they developed a two-stream convolutional neural network that can fuse RGB and depth data before classification, enabling the network to learn features from both domains. They conducted extensive experiments to evaluate the accuracy of their method and confirmed that it outperformed existing approaches.

Russakovsky et al. [3] presented a study that explores the advancements in object recognition and the challenges associated with collecting large-scale ground truth data. They highlighted key breakthroughs in the field of object recognition and provided a detailed analysis of the current state of the art. The study also compared the accuracy of computer vision algorithms with that of human accuracy. The authors summarized the most successful algorithms for object recognition based on their observations and analyzed the success and failure modes of these algorithms. The authors' work provides valuable insights into the current state of object recognition research, and their findings can guide the development of more effective and accurate algorithms in the future.

Krizhevsky, A., et al. [4] demonstrated that highly complex and deep convolutional neural networks can achieve remarkable performance on a challenging dataset using solely supervised learning without unsupervised training. Despite improving the network's size and training, their performance is still orders of magnitude inferior to that of the inferotemporal cortex of the human visual system, indicating that there is still a significant gap in the current state-of-the-art computer vision systems and the human visual system. Nevertheless, their findings are a significant contribution to the development of computer vision and demonstrate the remarkable potential of deep learning approaches.

In their study, Schwarz et al. [5] proposed a novel technique to improve object categorization and instant recognition by incorporating depth information. They rendered objects from a canonical perspective and used coloring in the depth channel to address existing problems in the field. To evaluate their approach, they utilized the Washington RGB-D objects dataset and employed a CNN that was trained on a large image dataset. Interestingly, the authors found that their technique performed well even when the dataset size was reduced. This suggests that their approach is robust and can still achieve high levels of accuracy even with limited training data. Moreover, their study opens up new possibilities for incorporating depth information in object recognition tasks and provides valuable insights into improving the performance of CNNs for instant recognition. The findings of their research have practical implications for the development of more effective object recognition systems in real-world applications.

III. EXISTING TECHNOLOGY

Animal monitoring systems are designed to track and observe animals, their movements, and behaviors. Various technologies have been developed by wildlife researchers for this purpose, including video monitoring systems mounted on animals, which can be intrusive and harmful (Parkhi et al., 2012). Other popular technologies include very high-frequency radio-tracking (Kim et al., 2010), GPS tracking (Tomkiewicz et al., 2010), satellite tracking with radio collars (Venkataraman et al., 2005), pyro-electric sensors (Hao et al., 2006), and wireless sensor networks (Mainwaring et al., 2002). However, these intrusive technologies have been largely limited to small geographic areas. With advances in camera trap technology (Wearn and Glover-Kapfer, 2017), non-intrusive methods for monitoring animals have become more reliable and effective. A comprehensive animal monitoring system must include both animal detection and classification. Various techniques have been used for animal detection, including animal motion (Dewan et al., 2007), threshold segmentation (Kitt et al., 2010), adaptive threshold segmentation (Chen, 2009), animal face detection (Burghardt and Calic, 2006), texture descriptor (Ramanan et al., 2005), and Content-Based Image Retrieval (CBIR) (Chen et al., 2005). However, threshold segmentation and motion-based detection methods only work on a static background, and adaptive threshold segmentation can produce many false detections by detecting other moving objects. Face recognition is also an inefficient technique for animal detection. Texture descriptors match the detected animal's texture with a predefined database, but this method only works well for a single type of animal and minimal background clutter. CBIR performance suffers when the database is enormous. With the advent of Convolutional Neural Networks (CNNs) (LeCun et al., 1999), the focus has shifted to 19 state-of-the-art machine learning models for animal detection and classification.

Animal breed classification

Fine-grained classification (FGC) is becoming increasingly important in the field of computer vision, especially in cases where the classes in a dataset have significant inter-class similarity and intra-class variance. One of the most prominent problems in FGC is animal breed classification, and several benchmark animal breed datasets have been developed to study this problem. These benchmark datasets include the Columbia Dogs with Parts (CU), Stanford dogs (SD), and Oxford IIIT-Pet dataset (OX). Researchers in the field of animal breed classification have used these benchmark datasets to evaluate the performance of various algorithms and approaches. The literature on animal breed classification is extensive and is categorized based on the benchmark datasets used in the research. Overall, FGC is a challenging problem that requires the development of specialized algorithms and techniques to achieve high accuracy and robustness in classification tasks. As more public datasets become available, the importance of FGC is likely to grow, and new research in this area will continue to contribute to the development of more effective and efficient computer vision systems.

Visible image-based systems

Camera traps have become an essential tool in wildlife research and conservation efforts, and they are widely used for animal detection and classification. Visible images have been the primary data source for camera trap-based animal detection and classification systems due to their richness in features such as color and texture. Convolutional neural networks (CNNs), which have achieved state-of-the-art performance in image classification tasks, are commonly used in these systems. These CNN architectures are often trained on large-scale visible image datasets, making them well-suited for camera trap-based animal detection and classification. Moreover, there are several benchmark camera trap datasets available that are based on visible images, providing researchers with ample opportunities to develop and evaluate their systems. The categorization of animal detection and classification systems based on the camera trap datasets used in the research can provide insights into the performance of these systems and guide the development of more effective approaches.

AlexNet Pre-Trained Convolutional Neural Network

Convolutional neural networks (CNNs) have proven to be a powerful tool in solving a wide range of visual recognition problems. While training an entire CNN from scratch can be computationally expensive and time-consuming, using pre-trained networks has become a common practice. Pre-trained networks have already learned a rich set of features that can be used for feature extraction and transfer learning. Popular pre-trained networks, such as

AlexNet, VGG-16, VGG-19, GoogleNET, and Inception-v3, have been trained on large datasets, containing millions of images classified into thousands of object categories. AlexNet, for instance, is a CNN that consists of eight layers, five of which are convolutional, and the other three are fully connected layers. To extract features using AlexNet, the first seven layers, including the five convolutional layers and the two fully connected layers, are considered, and the last fully connected layer is removed. The input images are usually resized to 227 x 227 x 3 to be compatible with AlexNet. The output of AlexNet is a 4096-dimensional feature vector for each image, which can be used as input to a linear classifier to classify the images. This technique of using pre-trained CNNs for feature extraction and transfer learning has been applied to a wide range of image recognition tasks and has been shown to produce remarkable results, especially when the training dataset is small.

LH1 animal-face dataset

Si and Zhu's (2011) animal-face dataset is a significant contribution to the field of animal classification, specifically in the fine-grained animal classification (FGC) category. The dataset comprises 19 classes of animal faces, each with significant inter-class and intra-class variability, making the classification task challenging. Additionally, the authors proposed a generative learning model that requires minimal training data for classification. This approach utilizes a template-based system, where each animal class has a template that learns various features, such as orientation, scale, texture, color, and local sketches. The baseline model achieved an impressive accuracy of 75.6%, considering the complexity and variability of the dataset. Si and Zhu's study demonstrates the potential of template-based models and the feasibility of training effective models with minimal training data, which is beneficial for practical applications where collecting large-scale annotated data can be challenging. The animal face dataset can be utilized for benchmarking and evaluating the performance of various FGC models.

Animal detection systems with their own dataset

There have been various research works to enhance the performance of animal detection and classification systems, which have practical applications in numerous fields. For instance, Maiti et al. (2015) utilized a camera trap dataset to investigate the abundance of grizzly bears in Canada's Banff National Park. They used the MSER technique to segment the bear from the images and then employed a CNN to classify the images that contain bears. The system achieved a recall rate of 63% and a precision of 7%. Meanwhile, Ragab et al. (2011) developed an innovative collision avoidance system for camel vehicles that employs animal detection and warning subsystems. By detecting the movement of camels on the road with the aid of a programmable GPU device, the driver can be alerted accordingly. These studies demonstrate the potential of animal detection and classification systems to improve safety for both animals and humans. However, the challenges encountered in these studies underscore the need for further research in this area.

Challenges

In this section, we list some of the challenges involved in animal classification and retrieval.

· Occlusion: Occlusion refers to the phenomenon
where the animals are partially or fully obstructed from view by objects such as trees, bushes, or other animals. This leads to incomplete or distorted information, making it challenging for the system to accurately identify the animal.
· Fast-Moving Nature: The fast-moving nature of animals makes it difficult to capture them in a clear and stable image, which results in motion blur and other artifacts that affect the quality of the image. Moreover, the sudden movement of animals also makes it challenging to track their movements accurately, leading to missed detections or false positives.
· Viewpoint variations: Viewpoint variations refer to changes in the orientation and position of animals in images or videos. These variations can lead to changes in the shape, pose, and rotation of the animal. These changes in viewpoint can cause difficulties for animal classification and retrieval systems, as they need to be able to recognize animals from different angles and orientations.
· Illumination: Light variation can affect the representation of animals. This can happen because of light falling from a different angle on animals.Different lighting conditions, such as varying intensity, direction, and color temperature, can result in variations in the texture, color, and contrast of the animal's image.
· Complex Background: Animals are often seen with greenery, covered with trees, soil, and water, which makes the classification task a difficult one.
· Intra-Class Variability and Inter-Class Similarity: Intra-Class Variability is the variations within the class and Inter-Class Similarity is the closeness across the different classes. Animals can be distinguished by using color, texture, and shape properties. These variations make it difficult in finding suitable features and also for the classifier to learn features used for better discrimination.

IV. PROPOSED METHODOLOGY

The animal detection and tracking model proposed in this study comprises several phases that operate in a sequential manner. Firstly, a selected video file is subjected to a file execution process that converts it into frames. The segmentation phase is then implemented to distinguish the objects of interest from the background regions present in the frames. The result is a series of segmented frames, which are subsequently inputted into the tracking algorithm to monitor and follow the animals across consecutive frames. Once the animals have been successfully tracked, the object detection module is activated to match the training images with the testing image, which is extracted and segmented to identify the objects. The output of this module labels the detected objects in all the successive frames of the video. A block diagram in Figure 1.1 visually depicts the different stages of the proposed model, including segmentation, tracking, and detection of animals.

The proposed model for animal detection and segmentation involves a simplified graph cut-based segmentation and a simplified maximum similarity-based region merging segmentation. The model consists of two key stages: identifying seed points and segmentation. Initially, the user provides seed points to specify the object and background region of interest. Following this, the rest of the segmentation process is automated, resulting in a final segmented output where animal regions are merged into a single region, and non-animal regions are segmented for easy removal. This approach simplifies the segmentation process and reduces the need for manual effort in identifying animal regions in an image. The proposed model has the potential to enhance the accuracy of animal detection and segmentation while also reducing processing time. A block diagram of the segmentation process is presented in Figure 1.1.

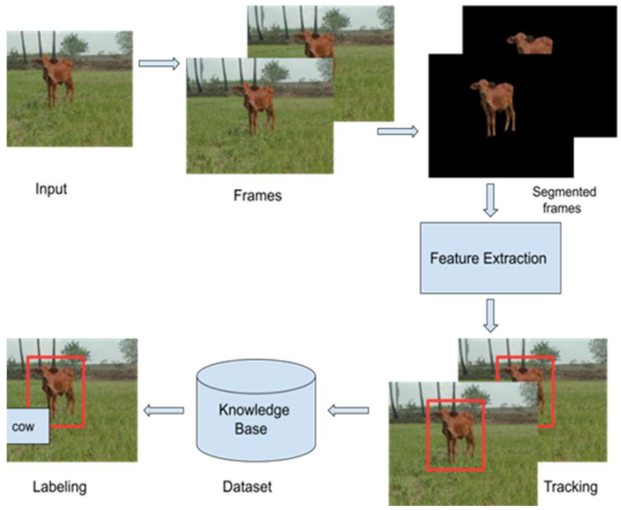

Figure 1.1 Block diagram of the proposed tracking model

Segmentation

The segmentation of animals in natural scenes is a challenging task due to the complexity of the background and the variability of animal appearance. Fully automatic segmentation algorithms may not always be effective due to the complex background, occlusion, and other factors that can interfere with accurate detection. Therefore, a semi-automatic segmentation algorithm that involves user interaction can be a more effective approach.The proposed model for animal segmentation involves a two-stage process: identifying seed points and segmentation. The user provides some seed points in the initial frame to indicate the foreground, which represents the animal, and the background, which represents the boundary of the frame. These seed points provide a starting point for the algorithm to identify the animal region and the non-animal region.In the second stage, the algorithm uses a maximal similarity-based region merging approach to merge the identified regions into the animal region and the non-animal region. This approach compares the similarity between different regions based on their features such as color, texture, and shape, and merges regions that are similar to each other.This semi-automatic segmentation approach provides a more accurate and efficient method for animal segmentation in natural scenes. By involving user interaction in the process, the algorithm can better account for the variability and complexity of the background, as well as the appearance of the animal itself.

Local Binary Pattern

The Local Binary Pattern (LBP) operator is a popular image operator that is used to convert an image into an array or image of integer labels that capture the small-scale appearance of the image. These labels, or their statistics, such as the histogram, are then utilized for further image analysis. Originally designed for monochrome still images, the LBP operator has been extended to work with color (multi-channel) images, videos, and volumetric data. One of the major advantages of LBP features is their versatility and high performance in a broad range of applications. For example, LBP features are widely used in texture classification and segmentation. They are highly effective at identifying and separating different textures within an image. In addition, LBP features have been employed in other applications, such as object recognition, face recognition, and image retrieval. Additionally, LBP features have proven to be highly effective in image retrieval and surface inspection applications, where they can quickly and accurately identify objects or features of interest.At its core, the original LBP operator works by labeling the pixels of an image. It does this by thresholding the 3-by-3 neighborhood of each pixel with the center pixel value, and then considering the resulting binary values as a single binary number. By applying this process to all pixels in the image, the LBP operator can generate a descriptor that effectively captures the texture and small-scale features of the image.

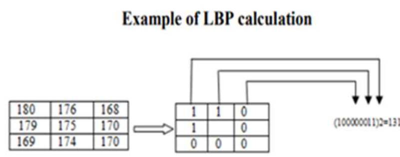

Fig 1.2 LBP calculation example

The Local Binary Pattern (LBP) operator is a versatile image operator with a range of applications, including texture classification and segmentation, image retrieval, and surface inspection. The operator generates an array or image of integer labels that represent the small-scale appearance of an image. These labels, or their statistics, such as the histogram, can then be used for further image analysis. The 256-bin histogram of the labels is a popular texture descriptor, and each bin can be considered a micro-texton. These micro-textons represent different types of local primitives, such as curved edges, spots, and flat areas, as shown in Figure 1.3. To accommodate different neighborhood sizes, the LBP operator has been extended. For example, the LBP4, 1 operator uses four neighbors, while LBP16, 2 considers 16 neighbors on a circle of radius 2. The operator LBPP, R generally refers to a neighborhood size of P equally spaced pixels on a circle of radius R that form a circularly symmetric neighbor set. This approach enables the LBP operator to be applied to various types of images, including color images, videos, and volumetric data.

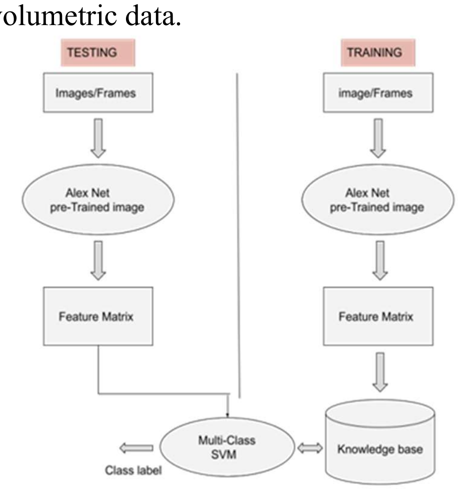

Fig 1.3 Testing and Training of images

In the proposed model, both the training and testing images undergo a resizing process, where they are resized to 227 x 227 pixels. This resizing process is done to maintain consistency in the size of images and to reduce the computational cost. The model is implemented on a 1GB Radeon HD 6470M GPU processor, which is a graphics processing unit specialized in handling large amounts of image data.

For the training phase, Alex Net pre-trained weights are used to train the images. The pre-trained weights provide a starting point for the model to learn from and extract features from the images. During the training phase, the model extracts 4096 features from each image.

Throughout the testing phase of image classification, an animal image that lacks any labeling is analyzed by extracting a consistent set of features through the implementation of a multi-class SVM classifier. In this instance, the linear kernel is utilized for SVM training, a widely used kernel function that is particularly effective for image classification tasks.

To evaluate the performance of the proposed model, experiments were conducted for five trials. The model's performance was evaluated using standard validity measures such as precision, recall, and F-measure, as discussed in Chapter 1. These measures are computed from a confusion matrix obtained during the classification of animal images and animal videos. The confusion matrix provides a breakdown of the number of true positives, true negatives, false positives, and false negatives. Based on these measures, the proposed model's accuracy and effectiveness in animal classification can be evaluated.

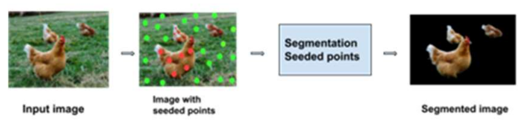

Fig 1.4 Segmentation of image

Acquisition

To acquire a diverse set of images for animal classification, images are captured in various environments with different backgrounds, lighting conditions, and angles. Additionally, images are also downloaded from various sources on the web to ensure a wide range of diversity. The images captured and downloaded contain animals with varying degrees of deformation, and with high inter and intra-class variations, meaning that different images of the same animal can look significantly different due to various factors such as age, gender, and species. By including these diverse images in the dataset, the animal classification system can learn to recognize animals under various conditions and improve its overall accuracy.

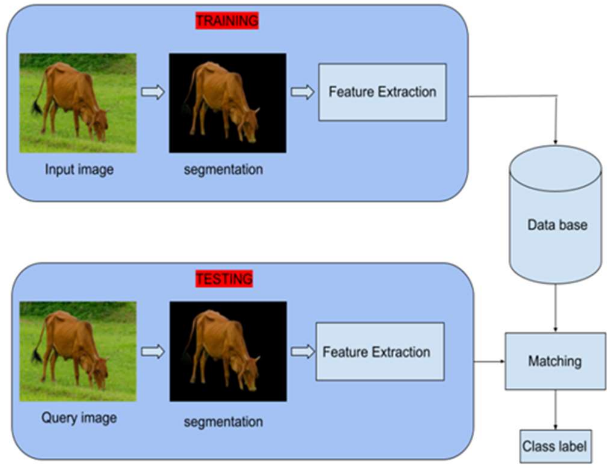

Fig 1.5 workflow of the model

Graph Cut-Based Segmentation

The Graph cut-based segmentation algorithm (Boykov & Jolly, 2001) is a highly efficient algorithm that enables the segmentation of animals in images. However, since the algorithm is interactive, it requires the input of seed points by the user to indicate the object and background regions, which are considered hard constraints. By computing the global optimum among all segmentations that satisfy the hard constraints, the algorithm can help segment animals automatically. To achieve this, a cost function that considers both boundary and region properties of segments as soft constraints is defined (Boykov & Jolly, 2001). The cost function involves the use of a binary vector that defines assignment to each pixel in P, which can be either an object, "obj", or a background, "bkg". This enables the algorithm to effectively segment animal images, even in the presence of complex backgrounds, different environmental conditions, and inter and intra-class variations.

Graph Cut Based Segmentation has an algorithm, that involves the following steps:

- Graph creation: tThe initial stage of graph-cut-based segmentation involves the generation of a graph that accurately represents the targeted image. This is achieved by allotting a node to each pixel within the image, functioning as the fundamental unit of computation. The interconnections between the pixels are subsequently depicted via edges, which are established based on pre-determined criteria such as proximity or intensity similarity. Through this graph structure, spatial and contextual information of the image can be effectively portrayed, allowing for precise and accurate segmentation to take place.
- Edge weighting: To assign weights to the edges in the graph, one common approach is to use pixel similarity or dissimilarity as a criterion. Specifically, for each pair of neighboring pixels, we can define a cost function that measures how much it would cost to assign them to different segments, taking into account their color, texture, position, or other features. A common choice is the pairwise Gaussian function, which assigns lower weights to edges connecting similar pixels and higher weights to edges connecting dissimilar pixels.The choice of edge weights can have a significant impact on the final segmentation result, as it balances the trade-off between fitting the data and

respecting the constraints. Ideally, we want to find the weights that maximize a global criterion, such as the energy function, that measures the total cost of the cut. However, this is generally an NP-hard problem, so we resort to approximate algorithms, such as graph cuts or belief propagation, to find good solutions efficiently.

- Source and sink selection: After constructing the graph and assigning weights to the edges based on the similarity between pixels or user-defined constraints, the next step is to designate a source node and a sink node. The source node represents the foreground or the object of interest, while the sink node represents the background or the parts of the image that are not of interest.Designating the source and sink nodes is a crucial step in the segmentation process, as it helps the algorithm to understand which parts of the image should be classified as objects and which parts should be classified as the background. Once the source and sink nodes have been identified, the graph cut algorithm calculates the minimum cost of separating the nodes connected by the edges, such that the source and sink nodes are in different segments.The process of designating the source and sink nodes can be done manually by the user, or automatically by using some predefined rules or algorithms.
- Minimal cut calculation: Once the graph has been constructed, and weights have been assigned to the edges, the next step is to designate the source and sink nodes, representing the foreground and background, respectively. This step is crucial in determining the segmentation of the image, as it defines the boundaries between the object of interest and the background. To achieve this, a minimum cut algorithm is utilized to determine the minimum cut separating the source and sink nodes while preserving graph connectivity. The minimum cut represents the optimal segmentation of the image, where the nodes on one side of the cut corresponding to the foreground (object of interest), and the nodes on the other side correspond to the background. The minimum cut algorithm works by finding the cut that minimizes the total weight of the edges crossing the cut, subject to the constraint that the source and sink nodes remain on opposite sides of the cut. This can be achieved through various algorithms, such as the Ford-Fulkerson algorithm or the Boykov-Kolmogorov algorithm, both of which have been shown to be effective in graph cut-based segmentation.
- Label updating: After obtaining the minimum cut separating the source and sink nodes, the graph cut-based segmentation algorithm proceeds to update the labels of the pixels in the image. This is done by assigning pixels connected to the source node to the foreground and those connected to the sink node to the background. Essentially, the minimum cut serves as a boundary between the foreground and background regions, which allows the segmentation of the image into two distinct regions. This labeling process involves modifying the binary vector that defines the assignment of pixels as either object or background. Pixels on one side of the cut are labeled as the foreground, and those on the other side are labeled as the background. By updating the pixel labels based on the minimum cut result, the algorithm can effectively separate the object from the background in the image.
- Repeat until convergence: In case the minimum cut result obtained in the previous step has changed, the algorithm goes back to the edge weight assignment step and updates

the weights to improve the separation of foreground and background regions. The process is repeated iteratively until the minimum cut value remains constant or until a maximum number of iterations has been reached. The maximum number of iterations is usually set to ensure that the algorithm does not run indefinitely and to prevent over-segmentation or under-segmentation of the image.

Region Merging

Once the initial segmentation is complete, there will be several small regions, which can be further processed using a region merging algorithm. In this process, the user marks some regions as objects and some as background regions, which results in three regions: the marked object region, the marked background region, and the unmarked region.

To perform the region merging process, it is necessary to derive some rules that can represent the regions and the merging process. A region can be represented by various descriptors, such as size, color, and texture. However, the shape may not be an adequate descriptor in this case since the image is divided into small-sized regions.

Therefore, other descriptors such as color and texture can be used to represent the regions. Color descriptors are based on color histograms, which represent the distribution of colors in the region. Texture descriptors are based on the statistical properties of the pixel intensities in the region, such as mean, variance, and co-occurrence matrices.

Once the regions are represented using these descriptors, the merging process can be performed based on some similarity measures between regions. The similarity measure can be based on a combination of color and texture descriptors, or on any one of these descriptors individually. The merging process can be done using a hierarchical clustering algorithm, which groups regions that are similar to each other.

The merging process can be repeated until the desired level of segmentation is achieved. The resulting regions can then be labeled foreground or background based on the user's markings.

Feature Extraction

In the animal classification system, the feature extraction stage plays a crucial role in discriminating animals from each other. The goal of feature extraction is to identify and extract relevant information from the images that can distinguish between different animal classes. Discriminating features should be selected in such a way that their values among the same class patterns should be close to each other and different from other class patterns.

Various features can be used to describe the characteristics of animals, such as color, shape, and texture features. Color features describe the distribution of colors in the image, and can be used to distinguish between animals with different coat colors, for example. Shape features capture the overall shape of the animal, and can be used to differentiate between animals with different body shapes, such as a giraffe and an elephant. Texture features describe the spatial arrangement of image pixels and can be used to differentiate between animals with different fur or skin textures.

It is essential to select the appropriate set of features that can capture the distinguishing characteristics of the animals being classified. The extracted features will be used in the subsequent stages of classification, and the performance of the system will depend heavily on the quality of the features selected. Therefore, it is important to carefully consider the selection of features and experiment with different feature extraction techniques to identify the most effective ones for the animal classification problem at hand.

Classification

It is essential to identify discriminating features of animals that can effectively separate them into distinct categories. These features should be carefully chosen such that their values among animals belonging to the same category are similar, while those among animals belonging to different categories are distinct. Color, shape, and texture are some of the most common features used in animal classification. Each of these features can be further divided into sub-features, and different combinations of features can be used to improve the performance of the classification system.

The classification process involves training a model using a dataset of labeled animal images. The model learns to associate the features of animals with their respective categories. Once the model is trained, it can be used to classify new, unseen animals by extracting their features and comparing them to the learned patterns. The classification process may involve different algorithms, such as k-nearest neighbors, decision trees, support vector machines, and neural networks, among others. Each algorithm has its strengths and weaknesses, and the choice of algorithm depends on the specific requirements of the classification task.

Tracking

Automated animal tracking systems have been developed to address the challenges associated with manual tracking. These systems rely on computer vision techniques and machine learning algorithms to track animals in videos. However, the success of automated animal tracking systems depends on the accuracy of the algorithms used and the quality of the input video. Low-quality cameras, complex backgrounds, and occlusion can all affect the accuracy of automated animal tracking systems. In addition, tracking multiple animals in a video can be challenging since it requires distinguishing between each animal and keeping track of their movements over time. Different illumination and fast-moving animals can also pose challenges for automated animal tracking systems. Therefore, researchers continue to develop and refine computer vision techniques and machine learning algorithms to improve the accuracy and robustness of automated animal tracking systems.

Principal Component Analysis

Principal Components Analysis (PCA) is a widely used variable-reduction technique that aims to highlight variation and uncover meaningful patterns within a given dataset. The fundamental concept behind PCA involves the conversion of a large set of variables into a smaller set of "artificial" variables referred to as "principal components". These principal components capture the majority of the variance within the original variables, as illustrated in Figure 2 [7] and [8]. To conduct a Principal Component Analysis, the following general steps are typically followed:

- To begin the Principal Component Analysis, the entire dataset is utilized, regardless of any class labels, and is comprised of d-dimensional samples.

- The next step in Principal Component Analysis is to calculate the d-dimensional mean vector, which involves computing the mean value for each dimension of the entire dataset.
- Once the d-dimensional mean vector has been calculated, the next step is to compute the scatter matrix, also known as the covariance matrix, of the entire dataset. The scatter matrix is a square matrix that displays the variance and covariance between each pair of variables in the dataset.
- After computing the scatter matrix, the next step is to calculate the eigenvectors (e1, e2, e3, e4, e5,...,ed) and corresponding eigenvalues (λ1, λ2, λ3, λ4, λ5,...,λd) of the scatter matrix. The eigenvectors represent the directions in the dataset where the variance is maximal, while the corresponding eigenvalues indicate the amount of variance present in each eigenvector direction.
- Sort the eigenvectors by decreasing eigenvalues and choose k eigenvectors with the largest eigenvalues to form a d × k dimensional matrix W (where every column represents an eigenvector).
- Once the eigenvectors and eigenvalues have been calculated, a d x k eigenvector matrix is generated. This matrix is then used to transform the original samples into a new subspace. This transformation is performed using the equation:
  y = WT × x
  Where x is a d x 1-dimensional vector representing one sample, W is the eigenvector matrix with dimensions d x k, and T denotes the matrix transpose operation. The output, y, is the transformed k x 1-dimensional sample residing in the new subspace.
- During the transformation of the samples into the new subspace, the objective is to maximize the retained variance of the data, which is achieved by selecting the k eigenvectors that correspond to the k largest eigenvalues. This ensures that the principal components which account for the majority of the variance in the original dataset are retained in the transformed subspace. By maximizing the variance of the projected data, we can effectively reduce the dimensionality of the dataset while still preserving the most significant information.
- The least square reconstruction error is minimized (minimizes mean squared distance between data point).

Linear Discriminant Analysis

The procedure for conducting a Linear Discriminant Analysis involves several general steps:

- Calculate the mean vectors for each class in the training data, with a dimension of d.
- Calculate the scatter matrices, including the within-class and between-class scatter matrix.
- Obtain the eigenvectors and eigenvalues from the scatter matrix.
- Choose k eigenvectors, with k being the number of dimensions in the new feature space (k≤d), that have the highest k eigenvalues.
- Transform the original samples into the new k-dimensional feature space using the transformation matrix formed from the selected eigenvectors.

- Use the new feature space for classification purposes, such as training a machine learning algorithm like logistic regression or decision tree.
- Compute the d dimensional mean vectors for the different classes from the dataset.
- Compute the scatter matrices.
- Compute the eigenvectors (e1, e2,...,ed) and corresponding eigenvalues (λ1, λ2,...,λd) for the scatter matrices.
- Sort the eigenvectors by decreasing eigenvalues and choose k eigenvectors with the largest eigenvalues to form a $d \times k$ dimensional matrix W (where every column represents an eigenvector).
- Use this $d \times k$ eigenvector matrix to transform the samples into the new subspace. This can be summarized by the matrix multiplication:

  $$Y = X \times W,$$
- where X is a $n \times d$ dimensional matrix representing the n samples, and Y are the transformed $n \times k$ dimensional samples in the new subspace.

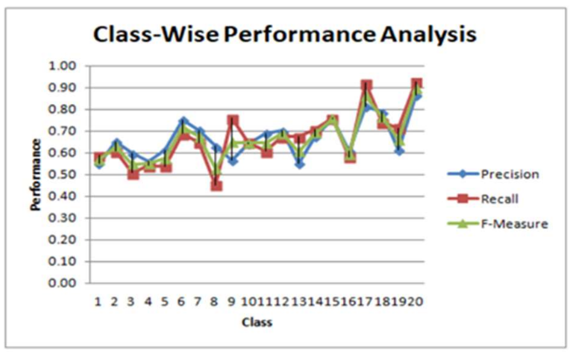

Fig 1.6 Class-wise performance of labeling in terms of precision, recall, and Fmeasure

V. CONCLUSION

In this paper, we propose two different approaches for the segmentation of animal images are proposed. The results are tabulated using different performance measures of segmentation by comparing both methods with the ground truth provided by human experts. The region merging-based segmentation is established to be more accurate and precise in segmenting animal regions. The different features like color, texture, and fusion of these features are extracted and classified using different classifiers like SVM and symbolic classifiers. Further, a system for detecting, segmenting, tracking, and labeling animals in videos is proposed. Finally, the convolutional neural network approach is explored to classify animals in both images and videos. The summary of each chapter is given below. Epilogue 93 In chapter 1, we present an overview of animal classification and tracking systems along with a general architecture of the animal classification system. A brief survey of existing related works on animal classification and tracking is presented. The challenges involved in the development of a classification and tracking system along with its application are listed. The performance measures used to evaluate the proposed segmentation and classification models are also presented.

The provided methodologies (PCA, LDA, SVM, and LBPH) will be compared with other current algorithms in future work through trials and tests of more complicated algorithms (deep learning). We also want to use larger animal picture datasets to test the validity of the methodologies that have been described. The performance of the classifier must then be enhanced by utilizing a mix of local descriptors. Future projects may involve testing this methodology on datasets of additional animals.

REFERENCES

[1] XIE, Z., A. SINGH, J. UANG, K. S. NARAYAN, and P. ABBEEL. Multimodal blending for high accuracy instance recognition. In: 2013 IEEE/RSJ International Conference on Intelligent Robots and Systems.

[2] Tokyo: IEEE, 2013, pp. 2214–2221. ISBN 978-1-4673-6356-3.
DOI: 10.1109/IROS.2013.6696666.

[3] EITEL, A., J. T. SPRINGENBERG, L. D. SPINELLO, M. RIEDMILLER and W. BURGARD. Multimodal Deep Learning for Robust RGB-D Object Recognition. In: 2015 IEEE/RSJ, International Conference on Intelligent Robots and Systems (IROS). Hamburg: IEEE, 2015, pp. 681– 687. ISBN 978-1-4799-9994- 1. DOI: 10.1109/IROS.2015.7353446.

[4] RUSSAKOVSKY, O., J. DENG, H. SU, J. KRAUSE, S. SATHEESH, S. MA, Z. HUANG, A. KARPATHY, A. KHOSLA, M. BERNSTEIN, A. C. BERG and L. FEI-FEI. ImageNet Large Scale Visual Recognition Challenge. International Journal of Computer Vision (IJCV). 2015, vol. 115, no. 3, pp. 211–252. ISSN 1573-1405. DOI: 10.1007/s11263-015-0816- y.

[5] KRIZHEVSKY, A., I. SUTSKEVER and G. E. HINTON. ImageNet classification with deep convolutional neural networks. Annual Conference on Neural Information Processing Systems (NIPS). Harrah's Lake Tahoe: Curran Associates, 2012, pp. 1097–1105. ISBN 978-1627480031Agarwal.

[6] RGB-D Object Dataset. In: the University of Washington [online]. 2014. Available at: http://rgbd-dataset.cs.washington.edu/dataset/

[7] SCHWARZ, M., H. SCHULZ and S. BEHNKE. RGB-D object recognition and pose estimation based on pre-trained convolutional neural network features. In: IEEE International Conference on Robotics and Automation (ICRA). Seattle: IEEE, 2015, pp. 1329–1335. ISBN 978-1-4799- 6923-4. DOI: 10.1109/ICRA.2015.

[8] HAWLEY, T., M. G. MADDEN, M. L. O'Connell and A. G. RYDER. The effect of principal component analysis on machine learning accuracy with high-dimensional spectral data. Knowledge-Based Systems. 2006, vol. 19, Iss. 5, pp. 363–370. ISSN 0950-7051.

[9] ABUROMMAN, A. A. and M. B. I. REAZ. Ensemble SVM classifiers based on PCA and LDA for IDS. In: 2016 International Conference on Advances in Electrical, Electronic and Systems Engineering (ICAEES). Putrajaya: IEEE, 2016, pp. 95– 99. ISBN 978-1-5090-2889-4. DOI: 10.1109/ICAEES.2016.7888016.

[10] HAGAR, A. A. M., M. A. M. ALSHEWIMY, and M. T. F. SAIDAHMED. A new object recognition framework based on PCA, LDA, and K- NN. In: 11th International Conference on Computer Engineering & Systems (ICCES). Cairo: IEEE, 2016, pp. 141—146. ISBN 978-1-5090-3267- 9. DOI: 10.1109/ICCES.2016.7821990.

[11] KAMENCAY, P.,T. TRNOVSZKY, M. BENCO, R. HUDEC, P. SYKORA and A. SATNIK. Accurate wild Animal recognition using PCA, LDA, and LBPH, In 2016 ELEKTRO. Strbske Pleso: IEEE, 2016, pp. 62–67. ISBN 978-1-4673-8698-2. DOI: 10.1109/ELEKTRO.2016.7512036

[12] STEKAS, N. and D. HEUVEL. Face Recognition Using Local Binary Patterns Histograms (LBPH) on an FPGA-Based System on Chip (SoC). In: 2016 IEEE International Parallel and Distributed Processing Symposium Workshops (IPDPSW). Chicago: IEEE, 2016, pp. 300–304. ISBN 978- 1-5090-3682-0. DOI: 10.1109/IPDPSW.2016.67.

[13] FARUQE, M.O. and A.M. HASAN. Face recognition using PCA and SVM. In: 2009 3rd International Conference on Anti-counterfeiting, Security, and Identification in Communication. Hong Kong: IEEE, 2009, pp. 97–101. ISBN 978-1-4244- 3883-9. DOI: 10.1109/ICASID.2009.5276938

[14] VAPNIK, V. Statistical Learning Theory. New York: John Wiley and Sons, 1998. ISBN 978-0- 471- 03003-4.

[15] WU, J. L. and W. Y. MA. A Deep Learning Framework for Coreference Resolution Based on Probabilistic neural network. In: 2017 IEEE 11th International Conference on Semantic Computing (ICSC). San Diego: IEEE, 2017, pp. 61–64. ISBN 978-1-5090-4284-5. DOI: 10.1109/ICSC.2017.57

[16] LeNet - Convolutional Neural Networks in Python. In: PyImage- Search [online]. 2016. Available at: http://www.pyimagesearch.com/2016/08/01/lenet convolutional-neural-network-in-python/.

[17] Understanding probabilistic neural networks. In: WildML [online]. 2015. Available at: http://www.wildml.com/2015/11/understandincconvolutional-neural-networks-for-NLP/.

[18] MURPHY, K. P. Machine Learning: A Probabilistic Perspective. Massachusetts: MIT University Press. 2012. ISBN 9780262018029.

[19] Afkham, H. M., Targhi, A. T., Eklundh, J.-O., & Pronobis, A. (2008). Joint visual vocabulary for animal classification. In Pattern Recognition, 2008. ICPR 2008. 19th International Conference on (pp. 1–4). IEEE.

[20] Ardovini, A., Cinque, L., & Sangineto, E. (2008). Identifying elephant photos by multi-curve matching. Pattern Recognition, 41(6), 1867–1877. Bai, X., Liu, W., & Tu, Z. (2009). Integrating contour and skeleton for shape classification. In Computer Vision Workshops (ICCV Workshops), 2009 IEEE 12th International Conference on (pp. 360–367). IEEE.

[21] K. Seetharaman and N. Palanivel "Texture characterization, representation, description, and classification based on full range Gaussian Markov random field model with Bayesian approach", International Journal of Image and Data Fusion, (Taylor & Francis) ISSN: 1947-9832 Volume 4, Issue 4, 2013, pages 342-362.

[22] Boykov, Y. Y., & Jolly, M.-P. (2001). Interactive graph cuts for optimal boundary & region segmentation of objects in ND images. In Computer Vision, 2001. ICCV 2001. Proceedings. Eighth IEEE International Conference on (Vol. 1, pp. 105–112). IEEE.

[23] Bremner, D., Demaine, E., Erickson, J., Iacono, J., Langerman, S., Morin, P., & Toussaint, G. (2005). Output-sensitive algorithms for computing nearest-neighbour decision boundaries. Discrete & Computational Geometry, 33(4), 593–604. Burghardt, T., & Ćalić, J. (2006).

[24] Analyzing animal behavior in wildlife videos using face detection and tracking. IEE Proceedings-Vision, Image and Signal Processing, 153(3), 305–312.

[25] Burghardt, T., & Calic, J. (2006). Real-time face detection and tracking of animals. In Neural Network Applications in Electrical Engineering, 2006. NATURAL 2006. 8th Seminar on (pp. 27–32). IEEE. Burney, A., & Syed, T. Q. (2016). Crowd Video Classification using Convolutional Neural Networks. In Frontiers of Information Technology (FIT), 2016 International Conference on (pp. 247–251). IEEE.

[26] N. Palanivel and M.Vasanthapriya, “Real-Time People Tracking and Counting System Using Kalman Filter”, International Journal of Engineering Applied Sciences and Technology, 2016 Vol. 1, Issue 6, ISSN No. 2455-2143, Pages 17-22.

[27] Dr.N. Palanivel, “PCA AND RF: AN AUTOMATIC PLANT LEAF DISEASE DETECTION USING TEXTURE, SHAPE AND COLOR FEATURES”, International Journal of Engineering Applied Sciences and Technology, 2017 Vol. 2, Issue 5, ISSN No. 2455-2143, Pages 120-125.

[28] Chan, T. H., Jia, K., Gao, S., Lu, J., Zeng, Z., & Ma, Y. (2015). PCANet: A Simple Deep Learning Baseline for Image Classification? IEEE Transactions on Image Processing, 24(12), 5017–5032. https://doi.org/10.1109/ TIP.2015.2475625

[29] Das, M., Manmatha, R., & Riseman, E. M. (1999). Indexing flower patent images using domain knowledge. IEEE Intelligent Systems and Their Applications, 14(5), 24–33.

[30] Duda, R. O., Hart, P. E., & Stork, D. G. (1973). Pattern classification (Vol. 2). Wiley New York.

[31] Dr.N. Palanivel, “Identification of Leaf disease using Fuzzy c-mean and kernal fuzzy c-mean and suggesting the pesticides”, International Journal of Advanced Research in Science, Engineering and Technology (IJARSET), Volume 2 Issue 2, May 2017.