iJRASET
International Journal For Research in
Applied Science and Engineering Technology

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 8 **Issue:** III **Month of publication:** March 2020

DOI: http://doi.org/10.22214/ijraset.2020.3014

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.177
Volume 8 Issue III Mar 2020- Available at www.ijraset.com

Detecting Crows on Sowed Crop Fields using Simplistic Image processing Techniques by OpenCV in comparison with TensorFlow Image Detection API

Mohammed Azam Sayeed[1], Noor Ayesha[2], Mohammed Akram Sayeed[3]
[1]Department of Computer sciences, SET Jain University Bangalore -562112

Abstract: This paper unveils attempts to develop an eco-friendly technique to reduce the loss of crops yield caused by crows keeping biodiversity in mind. Crows detection on sowed crop field are alternatives to static scarring techniques like scarecrow which are not ineffective as crows are smart birds also avoid harmful treatments that could damage biodiversity. Observation is made from Simplistic Image processing techniques provided by OpenCV and compared with pre-trained models provided TensorFlow object detection API, which can be used by the computer to find the corresponding coordinates and deploy effective and precise scarring techniques through IoT systems.
Keywords: Agricultural losses, Crows, Open CV, color-based segmentation, HSV colorspace, morphological operation, Tensorflow object detection API, protos, COCO pre-trained models.

I. THE NECESSITY TO DETECT BIRDS ON CROP FIELDS

[1]In India, nearly 65% of the people are directly or indirectly dependant on the agricultural sector for economic survival. The annual income of farmers is significantly influenced by the yield of the crops, which is continuously decreasing due to natural phenomena and poor technological advancement. However, particular attention should be paid to the damage caused by birds. While the exact measure of the loss in yield associated with birds is unknown, farmers integrate several traditional and conventional techniques to grow and store grains and fruits. Many of the used methods are ineffective and affect the environment in the long run. There is a need to develop alternative techniques, with experts in the fields of ornithology, agricultural sectors, and field visits, to avoid irreversible harm to the Indian biodiversity. In India, two-thirds of the population is directly dependent on the agricultural sector for their livelihood which contributes to the one-fifth part of the GDP of the country .The one-fifth population of the world is living on only 2.4% of the total land area of the world (Singh et al., 2002). In recent years crop productivity decreases in India due to the increase of the infestation of birds.

Birds generally create a negative impact on most of the agricultural activities and some activities attract birds as special feeding opportunities (Ormerod and Watkinson, 2003). The food of the birds is of mostly three types which depend on grains, seeds, fruits, green vegetation of the crop plants and grasses. The systematic procedure to develop a robust statistical method for assessing bird damage to the crop, particularly to fruits which provides accurate assessment data that can be used for scientific research and evaluation of bird management methods and devices (Saxton, 2006). On the other hand, to reduce the damage of farmers and grain storage activity, various bird-killing techniques are used such as chemical repellent, net, spike guards, traditional methods such as shooting the birds with the gunshot, making sound with help of crackers to scare birds (Subramanya, 1982). All these birds' management methods are less effective and cause great damage to the certain to threaten species and the migratory bird which produces the adverse effect on the conservation of biodiversity on a local, regional and global scale.

Research finding from [1] are as follows, Delphi Method was used to calculate the impact of birds on the damage of agricultural yield This analysis leads to the conclusion that 73% farmers considered that damage produced by birds is serious problem and 85% expressed the need of modern eco-friendly bird scaring techniques rather than traditional scarecrows. Most of the farmers expressed that this damage is dependent upon types of crops and seasons and it varies area to area while 40% of farmers considered that damage of crops and grains is about 35-60% and 25% farmers expressed that this damage is 0-35% in certain area (Fig 1) while 65% farmers consider that traditional techniques of bird scaring are not effective.

Question	Yes	No	Can't say
Are birds producing damage to the crops and grains?	73%	22%	5%
Is there a need of modern ecofreindly bird scaring techniques?	85%	2%	13%
Is damage percentage depending upon type of crops?	90%	8%	2%
Is damage percentage depends upon seasons?	93%	4%	3%

Fig. 1 [1] Response of the farmers to the survey on the percentage of damage of crops and grains due to birds from Silvassa (DNH, India) and Indapur (Pune, Maharashtra)

A. Impact of crows in Agricultural Losses

[1]Many crops are damaged by birds, with a little knowledge available of actual economic loss is done by House Crow. The great damage to the crop is noticed when they are in the mature stage by the Baya and Munias during the observations carried in Hyderabad, India and these birds with house crow can reduce the crop yield by more than 55% (R.L. Bruggers,1986).

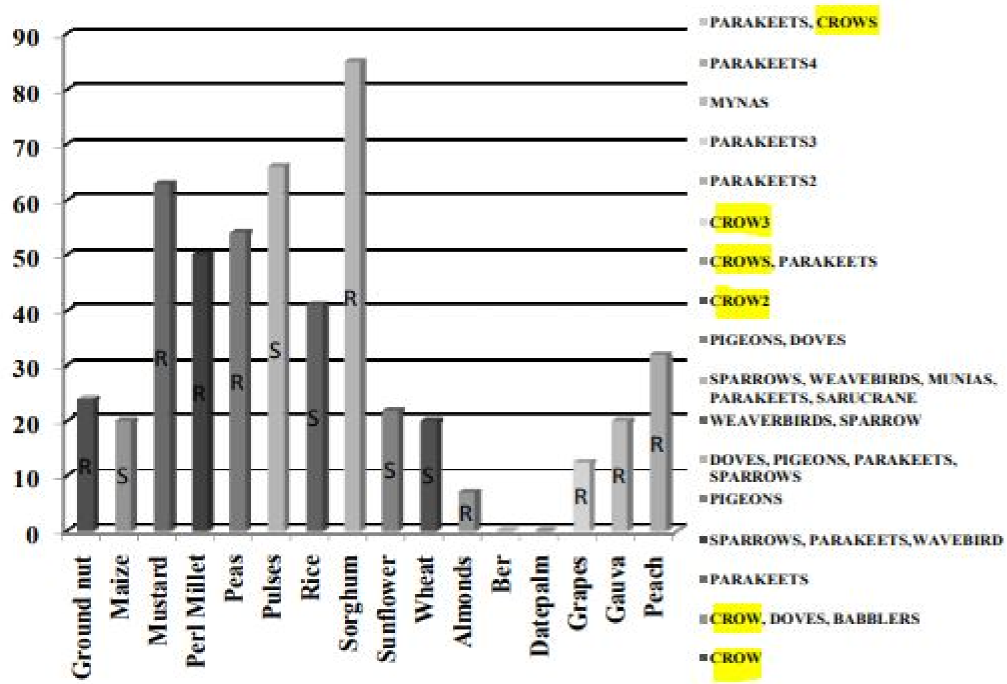

Fig. 2 [1]Extent of damage made by different bird species to crops and fruits (Perl Millet is recorded -10-100; Sorghum is recorded-12-85 in Ripening Season), RRipening Season and S-Sprouting (Dhindsa and Harjeet, 1994)

(Shivankar 2008), of Pune, Maharashtra, as his study area concludes that Pune and nearby area is known for its principal product of sugar cane and it is also known as 232 Manoj Kale et al sugar belt. The yield of sugar crop is affected by birds like House Crow because of their abundance and produce damage to the crop which is considerably large. This indicates that there is a need for the development of proper techniques that can save the birds and also the loss of farmers. Crows also observed damaging the crops of grapes to a great extent of grapes in Himachal Pradesh, India (Patyal and Rana, 2003). These damages can result not only limited to yield loss but also affects grapes which decrease the quality of the wine (Loinger et al.,1977). Thus bird pests constitute a significant limitation of productivity. According to the estimation of damage potential, it is found that loss due to bird is considerably large. Crows are also responsible for activities like spoiling the site area, damaging the gunny bags and contaminated grains with their droppings in the grain stores. Such research indicates that the need for research in damage estimation in Maharashtra, which is also the most important state in the field of agriculture and to develop the concerned measure.

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.177
Volume 8 Issue III Mar 2020- Available at www.ijraset.com

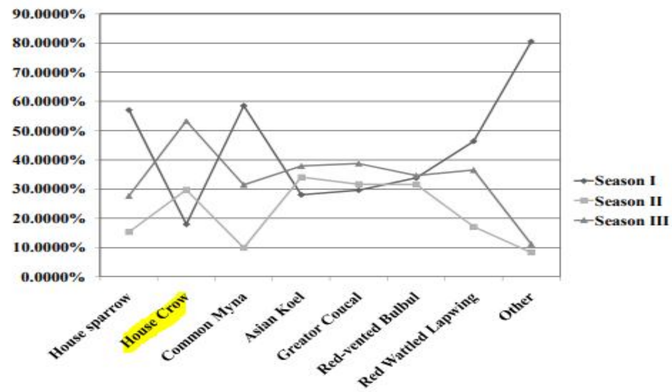

Fig. 3 [1] Relative abundance of birds in various seasons in sugarcane beet fields in Pune, Maharashtra, India. (Season I-Winter (2005-06); II-Summer (2006); III-Winter (2006-07)) (Data source: Shivankar, 2008)

B. Traditional static Scarring Techniques Inefficient for Crows

Frequency of complete eBird bird lists that report crows species as a rough indicator of abundance, especially in the south of India.

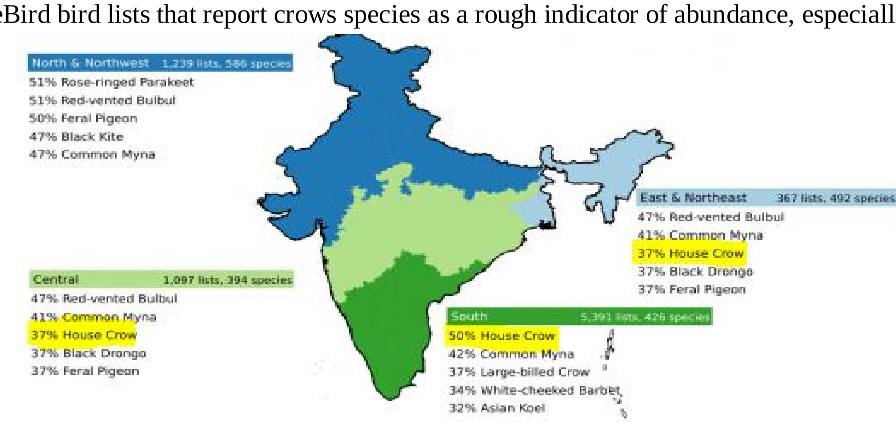

Fig. 4 Abundance of House Crow species in India

Scarecrows work quite well on some species of birds. Tests with scarecrows around small ponds reported a significant decrease in visits to those ponds by ducks. The crow is smarter than your average duck. Crows are some of the smartest birds in the avian world. They belong to the bird family Corvidae, sometimes referred to as the corvids. Crows are grassland walkers, systematically working their way through a field, Crows earned their role as pest birds because they can decimate a field crop by eating freshly scattered seeds, uprooting sprouted corn, picking ripe grapes from the vine, and pecking away at vegetables and grains before the harvest. Crows have adapted well to the rural and urban human landscape.

The traditional scarecrow is a decoy made in the shape of a human. It is dressed in old clothes and placed in an open field to discourage crows, blackbirds, and starlings from feeding on seeds and sprouting crops. The loose clothing flaps in the breeze, giving the illusion of a real, moving person. Stationary scarecrows may keep the crows away for a short period, but they quickly become used to them. Crows being a good problem solver, strategist and hunters in-group makes it necessary for farmers to depend on more intelligent systems to scarecrows which require a good detecting system of crows.

Few of the Alternative Methods are non-effective and most of the methods are not eco-friendly and produce direct harm to biodiversity. These techniques consist of mainly different bird repellent techniques such as visual repellent, chemical repellent, bioacoustics repellent, optical repellent. Killing most of the birds are illegal in India (Singh and Dungan, 1955) despite that it is strong belief among the farmers that Killing the birds is considered as the surest way to free from the problem of birds and they use techniques which consist of shooting, trapping, fumigation, poison baiting, egg, and nest destruction, killing with the help of gun and catching them in trap. All these will produce damage to the threat and migratory birds and also produce damage to the conservation of Damage to the Agricultural Yield 233 biodiversity.

Thus we need better automated systems to detect crows by computer systems to deploy IOT based scarring techniques that can effectively reduce the attack of crows to crop fields and agricultural losses, and also ensuring the safety of biodiversity and Environment.

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.177
Volume 8 Issue III Mar 2020- Available at www.ijraset.com

II. DETECTION OF CROWS ON FIELDS USING SIMPLISTIC IMAGE PROCESSING BY OPENCV

A. Image Acquisition

Images are processed in jpeg format. JPEG is a commonly used method of lossy compression for digital images, particularly for those images produced by digital photography. The degree of compression can be adjusted, allowing a selectable tradeoff between storage size and image quality. We are using misc.imread function from scipy package to read images alternatively imageio.imread can be used as this is not deprecated.

```
Out[112]: <matplotlib.image.AxesImage at 0x22abc95a438>
```

Fig. 5 Input image of crows on sowed field

B. Converting to HSV Colorspace

Originally the image is RGB colorspace, R, G, and B components of an object's color in a digital image are all correlated with the amount of light hitting the object, and therefore with each other, image descriptions in terms of those components make object discrimination difficult. For easier object discrimination we convert the image to HSV colorspace. HSV color space describes colors in terms of Hue, Saturation, and Value. In situations where color description plays an integral role, the HSV color model is often preferred over the RGB model. The HSV model describes colors similarly to how the human eye tends to perceive color. RGB defines color in terms of a combination of primary colors, whereas, HSV describes color using more familiar comparisons such as color, vibrancy, and brightness.

Explanation of HSV colorspace

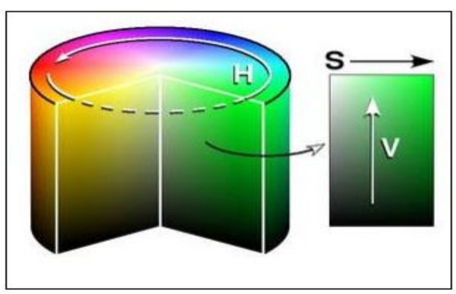

Fig. 6 HSV color wheel

Figure 6 illustrates how hue, saturation, and value are defined.

1) Hue represents the color type. It can be described in terms of an angle on the above circle. Although a circle contains 360 degrees of rotation, the hue value is normalized to a range from 0 to 255, with 0 being red.
2) Saturation represents the vibrancy of the color. Its value ranges from 0 to 255. The lower the saturation value, the grayer is present in the color, causing it to appear faded.
3) The value represents the brightness of the color. It ranges from 0 to 255, with 0 being completely dark and 255 being fully bright.
4) White has an HSV value of 0-255, 0-255, 255. Black has an HSV value of 0-255, 0-255, 0. The dominant description for black and white is the term, value. The hue and saturation level do not make a difference when the value is at max or min intensity level.

ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.177
Volume 8 Issue III Mar 2020- Available at www.ijraset.com

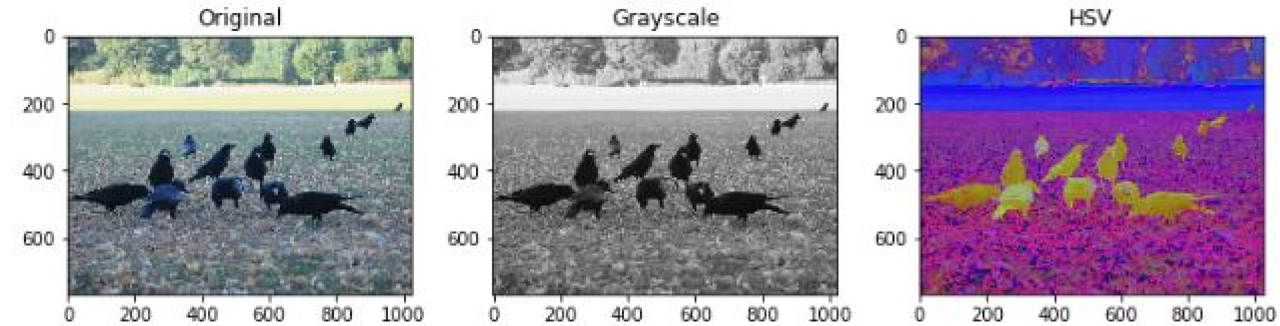

Fig. 7 Converted Colorspace representation of input image

As expected the HSV colorspace detects foreground in our case the crows against the background pixels more clearly in comparison to other colorspaces such as RGB or grayscale image , which makes it easier for further image processing.

C. Simplistic Segmentation Using Color Spaces

This is one of the most simplistic segmentation provided by the OpenCV library in python. This is a great choice for our paper as the birds of particular species in our case house crows have a specific range of color shades, throughout India. The key Python packages you'll need are NumPy, the foremost package for scientific computing in Python, Matplotlib, a plotting library, and OpenCV.

The HSV image is converted from RGB, using cvtColor() from OpenCV is used, then we need to pick the right HSV colorspace range to detect an object from the background. From Experimentation we found the below lower range array [0,0,0] and higher range [50,50,100] to be suitable for our application to detect crows from HSV color space image. The upper and lower shade can be visualized in matplotlib in corresponding RGB shade using the hsv_to_rgb function as given below in the figure.

```

hsv = cv2.cvtColor(img, cv2.COLOR_BGR2HSV)

lower_gray = np.array([0,0,0], np.uint8)
upper_gray = np.array([50,50,100], np.uint8)

mask_gray = cv2.inRange(img, lower_gray, upper_gray)
img_res = cv2.bitwise_and(img, img, mask = mask_gray)
```

```
from matplotlib.colors import hsv_to_rgb
lo_square = np.full((10, 10, 3), lower_gray, dtype=np.uint8) / 255.0
do_square = np.full((10, 10, 3), upper_gray, dtype=np.uint8) / 255.0
plt.subplot(1, 2, 1)
plt.imshow(hsv_to_rgb(do_square))
plt.subplot(1, 2, 2)
plt.imshow(hsv_to_rgb(lo_square))
plt.show()
```

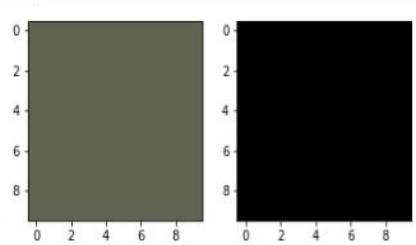

Fig. 8 Color-based Segmentation on HSV image

Once we have determined the upper and lower range of HSV color space range, we used cv2.inRange() to try to threshold crows. inRange() takes three parameters: the image, the lower range, and the higher range. It returns a binary mask (a ndarray of 1s and 0s) the size of the image where values of 1 indicate values within the range and zero indicate values outside.

mask_gray variable represents the output image of the inRange() function and img_res represented the RGB segmented image with actual pixel values and black pixels background.

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.177
Volume 8 Issue III Mar 2020- Available at www.ijraset.com

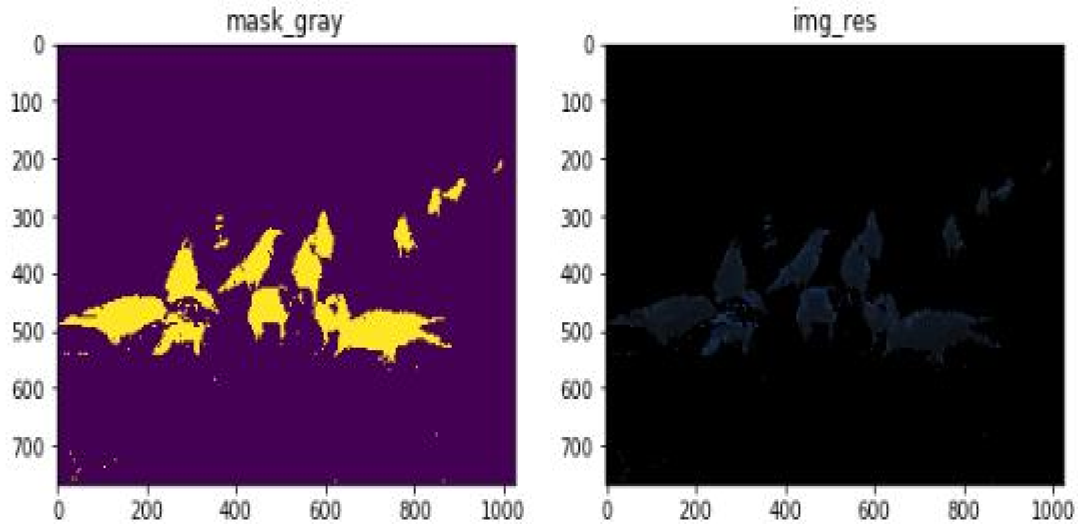

Fig. 9 mask from color-based segmentation and corresponding RGB segmented image

Observations can be made from Image obtained from inRange() as mask_gray is that the segmentation is a good approach for our problem, but it would need additional image processing to reduce the noise as it has retained small blobs in background.

Also the corresponding segmented RGB image img_res with background pixels as zero, clearly depicts segmentation is functionally correct but we need alternatives detection techniques to improve the comprehensibility of the image from human perspective which will be covered in the next steps.

D. Noise Reduction and Morphological Operations

We need to remove any small white noises in the image. For that, we can use the morphological opening. To remove any small holes in the object, we can use morphological closing. So, now we know for sure that region near to the center of objects is foreground and region much away from the object are backgrounds. The only region we are not sure of is the boundary region of crows. So we need to extract the area in which we are sure they are crows. Erosion removes the boundary pixels. So whatever remaining, we can be sure, it is the crow. That would work if objects were not touching each other. But since they are touching each other, another good option would be to find the distance transform and apply a proper threshold. Next, we need to find the area in which we are sure they are not coins. For that, we dilate the result. Dilation increases object boundaries to the background. This way, we can make sure whatever region in the background in the result is a background since the boundary region is removed.

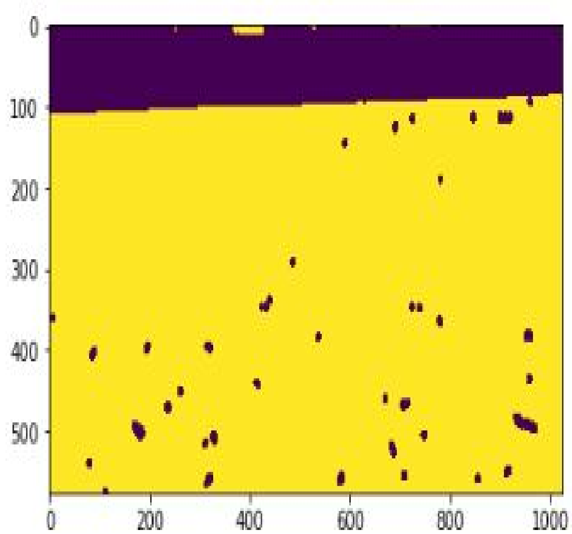

Fig. 10 Yellow region is sure background

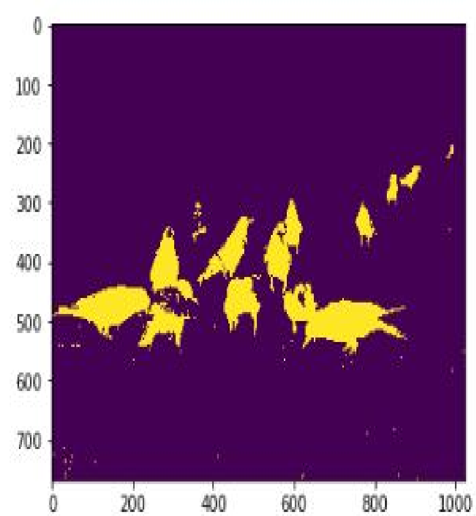

Fig. 11 yellow region is sure foreground

The code snippet is as below

```
# noise removal
kernel = np.ones((3,3),np.uint8)
opening = cv2.morphologyEx(thresh,cv2.MORPH_OPEN,kernel, iterations = 2)
# sure background area
sure_bg = cv2.dilate(opening,kernel,iterations=3)
# Finding sure foreground area
dist_transform = cv2.distanceTransform(opening,cv2.DIST_L2,5)
ret, sure_fg = cv2.threshold(dist_transform,0.7*dist_transform.max(),255,0)

sure_fg = np.uint8(fg)
unknown = cv2.subtract(bg,sure_fg)
```

Fig 12. Code snippet for noise removal and morphological opening

Now we know for sure which are the region of crows, which are background. So we create a marker (it is an array of the same size as that of the original image but with int32 data type) and label the regions inside it. The regions we know for sure (whether foreground or background) are labeled with any positive integers, but different integers and the area we don't know for sure are just left as zero. For this we use cv2.connectedComponents(). It labels the background of the image with 0, then other objects are labeled with integers starting from 1. But we know that if the background is marked with 0, watershed will consider it as unknown area. So we want to mark it with a different integer. Instead, we will mark the unknown region, defined by the unknown, with 0. Now our markers are ready. It is time for the final step, apply watershed. Then the marker image will be modified. The boundary region will be marked with -1.

```
# Marker labeling
ret, markers = cv2.connectedComponents(sure_fg,connectivity=4)
# Add one to all labels so that sure background is not 0, but 1
markers = markers+1
# Now, mark the region of the unknown with zero
markers[unknown==255] = 0
```

Fig. 13 Code snippet for watershed segmentation and marking

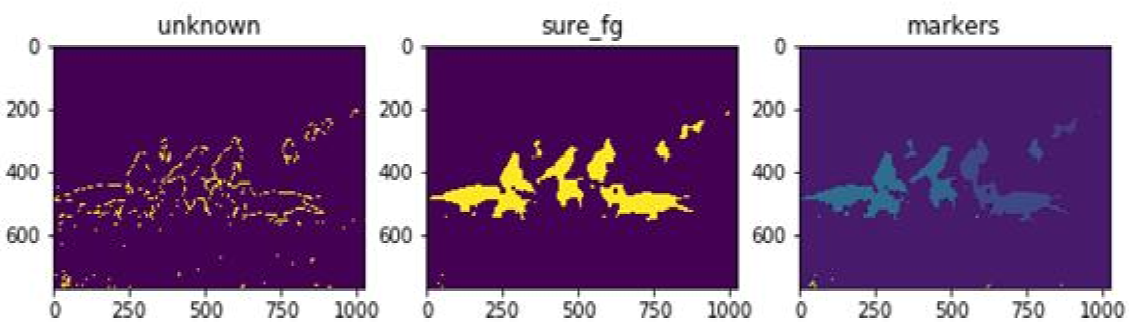

Fig. 14 Representation of above code snippet variables

Definition from Wikipedia: watershed is a transformation defined on a grayscale image. The name refers metaphorically to a geological watershed, or drainage divide, which separates adjacent drainage basins. The watershed transformation treats the image it operates upon like a topographic map, with the brightness of each point representing its height, and finds the lines that run along the tops of ridges.

There are different technical definitions of a watershed. In graphs, watershed lines may be defined on the nodes, on the edges, or hybrid lines on both nodes and edges. Watersheds may also be defined in the continuous domain. There are also many different algorithms to compute watersheds. The watershed algorithm is used in image processing primarily for segmentation purposes.

E. Markers on Actual Image for detection of Crows

The step will involve highlighting the Crows as Detected objects which can be used by the computer to find the coordinates of crows to deploy various and specific scarring technique which is not harmful to the environment and much more effective than static scarring techniques such as scarecrows etc.

Fig. 15 Final Image detection

The methodology works very well to Detect Crows from Blur Input images also as well as below

Fig. 16 Input Blur Image

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.177
Volume 8 Issue III Mar 2020- Available at www.ijraset.com

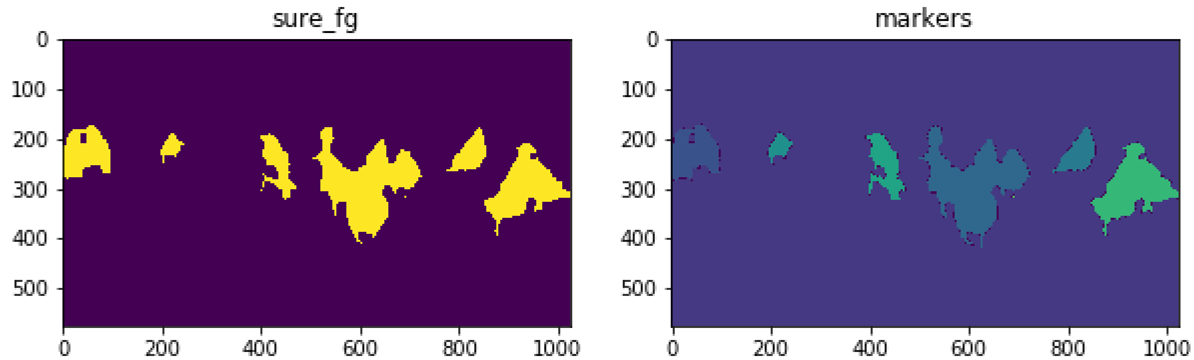

Fig. 17 Markers based on morphological operations

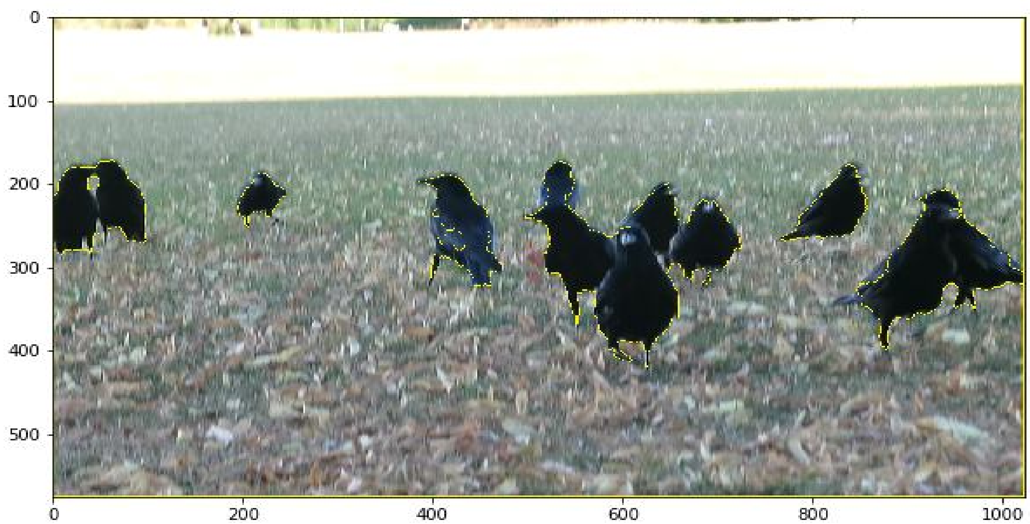

Fig. 18 Final Image detection of the blur input image

III.DETECTION OF CROWS ON FIELDS USING TENSORFLOW IMAGE CLASSIFICATION API

Tensorflow is Google's Open Source Machine Learning Framework for dataflow programming across a range of tasks. Nodes in the graph represent mathematical operations, while the graph edges represent the multidimensional data arrays (tensors) communicated between them. Tensors are just multidimensional arrays, an extension of 2-dimensional tables to data with a higher dimension. There are many features of Tensorflow which makes it appropriate for Deep Learning. So, without wasting any time, let's see how we can implement Object Detection using Tensorflow.

A. Environment Setup and Installation

1) Necessary Libraries

We need to download Tensorflow using the below pip command

For CPU

pip install tensorflow

and ensure these libraries are available

pip install --user Cython

pip install --user contextlib2

pip install --user pillow

pip install --user lxml

pip install --user jupyter

pip install --user matplotlib

and ensure these libraries are available, we can alternatively use Conda commands for Anaconda Environment.

2) Download of TensorFlow's Model

Github link: https://github.com/tensorflow/models
Rename the downloaded file as Model for simplicity, Save it in Tensorflow Folder preferably C: Drive, also ensure that the downloaded file have protos file in object detection model in research folder as below
C:\Tensorflow\models\research\object_detection\protos
Notice the many protos files available, but they are not readable and usable for us, we need step 3) for converting protos files to .py files

3) Protos Buffer installation

we have Protobuf: Protocol Buffers (Protobuf) are Google's language-neutral, platform-neutral, extensible mechanism for serializing structured data, – think of it as XML, but smaller, faster, and simpler. Download above v3.4 in the same folder hierarchy as downloaded models
Link: https://github.com/protocolbuffers/protobuf/releases
Folder structure: C:\Tensorflow\protoc-3.11.4-win32

4) Convert Protos files to .py files using the command as below
"path_of_protobuf's bin"./bin/protoc object_detection/protos/ (using exe file present in bin folder)
In our case, the command would be
C:/Tensorflow/Protoc-3.11.4-win32/bin/protoc object_detection/protos/*.proto –python_out=.
go to the protos folder inside models>object_detection>protos and there you can see that for every proto file there's one python file created.
Also, Install TensorFlow detection API using the below command
pip install tensorflow-object-detection-api

B. Tensorflow Object Detection API

Open the .py file generated from protoc command named as object_detection_tutorial.ipynb
Folder structure : C:\Tensorflow\models\research\object_detection\ object_detection_tutorial.ipynb
The API uses the Tensorflow detection model zoo to provide a collection of detection models pre-trained on the COCO dataset, the Kitti dataset, the Open Images dataset, the AVA v2.1 dataset, and the iNaturalist Species Detection Dataset. These models can be useful for out-of-the-box inference if you are interested in categories already in those datasets. They are also useful for initializing your models when training on novel datasets.

```
def load_model(model_name):
  base_url = 'http://download.tensorflow.org/models/object_detection/'
  model_file = model_name + '.tar.gz'
  model_dir = tf.keras.utils.get_file(
    fname=model_name,
    origin=base_url + model_file,
    untar=True)
  model_dir = pathlib.Path(model_dir)/"saved_model"
  model = tf.saved_model.load(str(model_dir))
  model = model.signatures['serving_default']
  return model


model_name = 'ssd_mobilenet_v1_coco_2017_11_17'
detection_model = load_model(model_name)
```

Fig. 19 Prebuild models used by TensorFlow object detection API

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.177
Volume 8 Issue III Mar 2020- Available at www.ijraset.com

From the Code in the API we find that COCO model is used for object detection, COCO stands for Common Objects in Context, this dataset contains around 330K labeled images. Now the model selection is important as you need to make an important tradeoff between Speed and Accuracy. Depending upon your requirement and the system memory, the correct model must be selected.Inside "models>research>object_detection>g3doc>detection_model_zoo" contains all the models with different speed and accuracy(mAP). Provide Input Images using the below variables , TEST_IMAGE_PATHS will have the folder structure containg the required input images

```
PATH_TO_TEST_IMAGES_DIR = pathlib.Path('C:\\Users\\mosa0816\\Project Urine\\Crow detection')
TEST_IMAGE_PATHS = sorted(list(PATH_TO_TEST_IMAGES_DIR2.glob("*.jpg")))

for image_path in TEST_IMAGE_PATHS:
  show_inference(masking_model, image_path)
```

Fig. 20 Variables used by TensorFlow API as input images

C. Results Obtained from Prebuild Models from TensorFlow API

Tensorflow object detection API can detect Crows as generic birds, but also misclassify few crows as dog and Sheep as in Fig 20. Also for blur image as in Fig 21, it was able to detect most of the Birds, few were not detected but were able to detect even far off objects like Person. Thus Tensorflow API is good for generic object detection but compromises on accuracy and precision of detecting specific objects like crows.

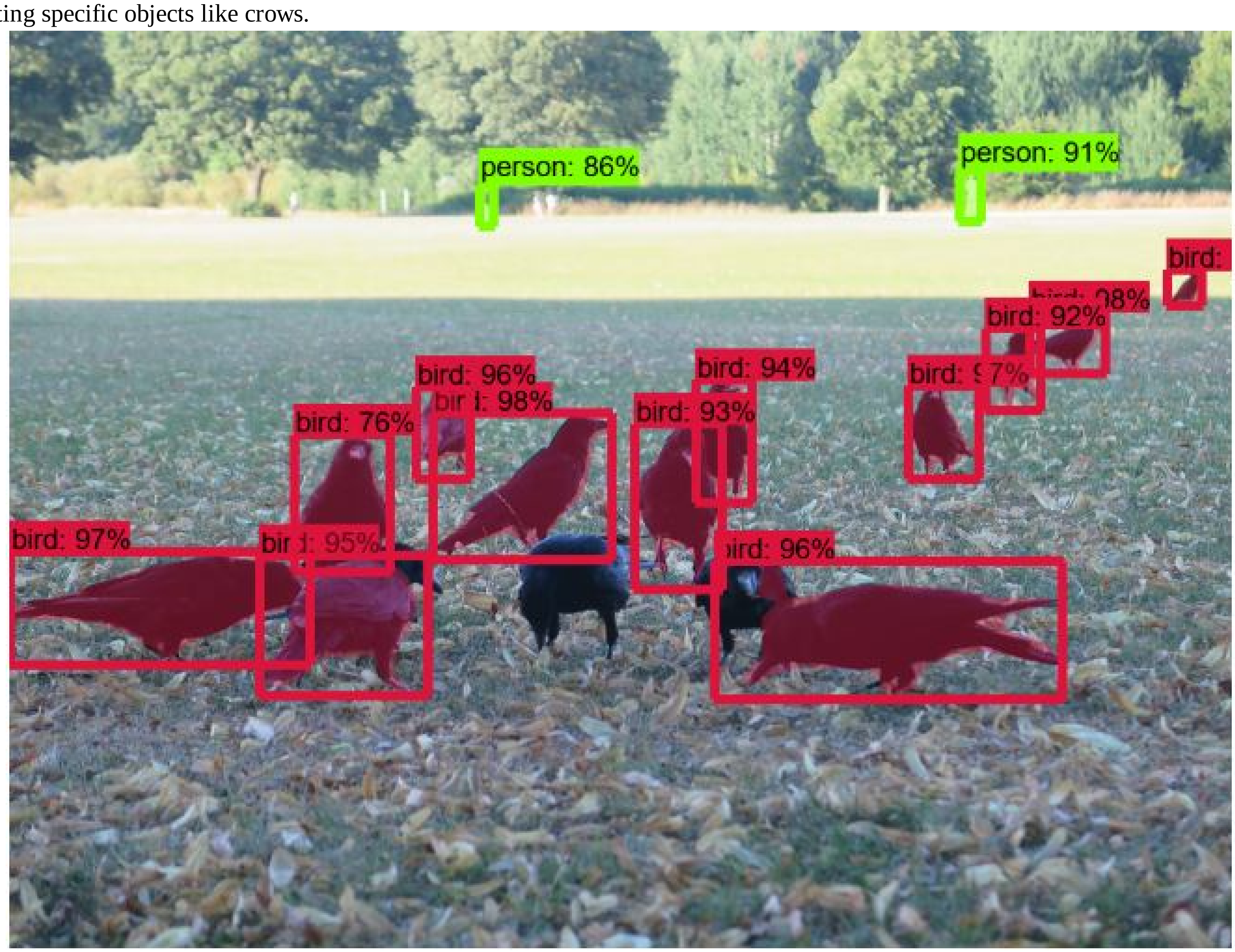

Fig. 20 Result for Input Image same as in Fig 5

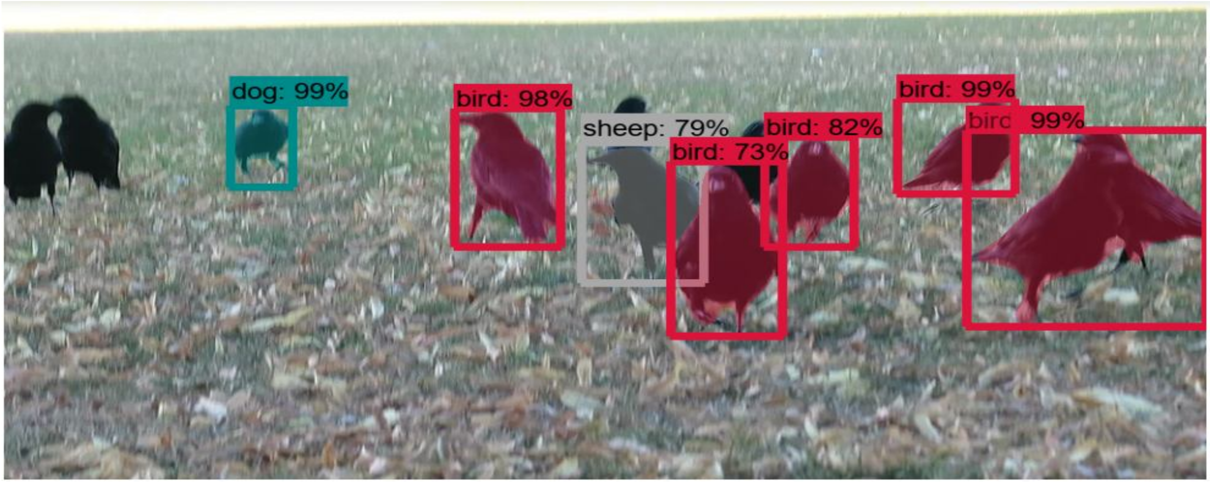

Fig. 21 Result for Blur Image same as in Fig 16

IV. COMPARISON OF SIMPLISTIC IMAGE PROCESSING TECHNIQUE TO TENSOR FLOW OBJECT DETECTION API

TABLE 1

COMPARISON OF SIMPLISTIC IMAGE SEGMENTATION TO TENSORFLOW OBJECT DETECTION API

Brand Name	OpenCV	Tensorflow object detection API
Functionality	OpenCV is a great computer vision library, all the algorithms, processing techniques are available. You can even accelerate OpenCV logic with Cuda support.	Tensorflow a graph computation library, with most people using it to construct deep neural networks. It can be used through C++ and Python interfaces on Linux and Android.
Documentation	Documentation is good with lots of examples available in Python, C/C++, android and ios as well	Documentation is mostly provided as Github repositories which may not be as readable or understandable as OpenCV documentation.
Use Cases	OpenCV is the widely used Computer Vision and Machine Learning toolbox, i.e. a set of methods and classes which can be used to build applications involving these two domains. It can easily be used in python for image processing.	Tensorflow has a different use case, it's a graph computation library, with most people using it to construct deep neural networks. It can be used through C++ and Python interfaces on Linux and Android. You can write and train such deep networks to perform a lot of Computer Vision tasks (such as human detection), but you would need a large amount of data and expertise in Deep learning (or a trained model and code).
Installation	It is very easy to install Opencv in python using corresponding pip commands, it is much more convenient for conda environments	Installation of Tensorflow is difficult to flow, with multiple GitHub repository downloads, and not much documented to follow through steps
Performance	It takes time and expertise to devise the image processing algorithm, but the well-formulated algorithm is extremely accurate, fast and reliable.	Installation takes time, API uses pre-trained models which is consumes large memory and processing power, also takes much more time compared to image processing algorithms.
Application	Opencv is advisable to use when we have a well-defined problem statement for specific object detection with a consistent background. But doesn't work well for generic object detection.	API works very well for classifying and detecting generic objects due to its large pre-trained models created from a large number of the trained image set. But it compromises on speed and performance.
Blur Images	We have great image pre-processing and segmentation techniques provided by OpenCV which can handle Blur images well.	Blur images can be missed by detection due to a lack of image preprocessing techniques on input images.
Conclusion	Opencv is better for devising an algorithm for specific object detection, giving faster and more accurate results. But may need assistance from ML techniques for complex and generic object detection scenarios.	Tensorflow object detection API work well for generic object detection scenarios because of its pre-trained models, but compromises on speed and performances. This is a great approach for unknown generic object detection scenarios but not for known image processing scenarios.

V. APPLICATIONS OF METHODOLOGY

The proposed system is capable of detecting Crows feeding on sowed crop fields which can be used by computers/humans for deploying effective scarring techniques which will not impact on biodiversity or environment. This paper also presents the observation from simplistic image processing algorithms in OpenCV with the comparison with state of the art TensorFlow Object detection API for object detection which can be used for the corresponding use case and requirements. The algorithm can be used by farmers to reduce agricultural losses and lays a foundation on this topic for researchers.

REFERENCES

[1] Manoj Kale, Berit Balfors, Ulla Mörtberg, Prosun Bhattacharya1, and Sanjay Chakane, "Damage to the Agricultural Yield due to Birds, Present Repelling Techniques and its Impacts: An Insight from the Indian Perspective" International Journal of Applied Agricultural Research, ISSN 0973-2683 Volume 6 Number 3 (2011) pp. 223-236, http://www.ripublication.com/ijaar.htm

[2] Mohammed Azam Sayeed & Shashikala G2, Sakshi Pandey, Ruhi Jain & Satish Kumar N," Estimation of Nitrogen in Rice Plant Using Image Processing and Artificial Neural Networks", Vol-2, Issue-8, 2016 ISSN: 2454-1362, http://www.onlinejournal.in,IJIR

[3] https://realpython.com/python-opencv-color-spaces/

[4] https://docs.opencv.org/3.3.1/d3/db4/tutorial_py_watershed.html

[5] https://www.edureka.co/blog/tensorflow-object-detection-tutorial/