iJRASET
International Journal For Research in Applied Science and Engineering Technology

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 10 **Issue:** IX **Month of publication:** September 2022

DOI: https://doi.org/10.22214/ijraset.2022.46793

www.ijraset.com

Call: 08813907089 | E-mail ID: ijraset@gmail.com

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 10 Issue IX Sep 2022- Available at www.ijraset.com

Camouflage Clothes Detection

Kondala Sreedattu[1], Prof. D. Lalitha Bhaskari[2]
[1]*Department of Computer Science & Systems Engineering, Andhra University College of Engineering(A), Andhra University, Visakhapatnam.*
[2]*Professor Department of Computer Science & Systems Engineering, Coordinator IQAC, Andhra University College of Engineering(A), Visakhapatnam.*

Abstract: Presentation of a study and implementation on task named camouflage clothes detection. The aim of the study is to identify clothes that "seamlessly" blend in with their surroundings.
Usage of a dataset, called 8k_normal_vs_camouflage_clothes_images, which consists of about 8k images of camouflage clothes and 8k images of normal clothes. Residual Networks (ResNets) are used for this task. The high intrinsic similarities between background and object make camouflage detection a challenging task.
Keywords: Pattern recognition, Artificial Neural Networks, Machine Learning, Image Analysis, Deep Learning.

I. INTRODUCTION

Background matching property of any object is called as camouflage. Some examples of camouflage include leopard's spotted coat, the battledress of a solider and so on [1]. Camouflage images can be split into two types natural camouflage and artificial camouflage. Natural camouflage is used by animals (e.g.., insects, cephalopods) as a tactic to avoid detection by a predator. Artificial camouflage usually occurs in product manufacturing so called defects and or used to hide information in art or gaming [2].

II. REALTED WORK

Two remarkable studies on camouflaged animals by Abbott Thayer [3] and Hugh Cott [4] are still prominent. The field of machine learning has seen huge rise by the introduction of Artificial Neural Networks (ANN). A variation of an ANN is Convolution Neural Networks (CNN). CNNs are used to solve difficult image driven pattern recognition tasks. The basic structure of an ANN is as shown in Figure 1. We load the input generally as a multi-dimensional vector to a first layer called input layer. The input layer then distributes to the hidden layer [5]. The hidden layer then makes some decisions based on the previous layer and weighs up how little changes contribute to the output [6].
These Artificial Neural Networks (ANN) are inspired from the way biological nervous systems operate (human brain) operate. ANNs are made up of many interconnected nodes just like the neurons. In Figure 1 a simple feedforward neural network (FNN) is shown. It consists of three layers an input layer, hidden layer, output layer. It is the basis for many neural networks such as Restricted Boltzmann Machines (RBMs), Recurrent Neural Networks (RNNs) [7].

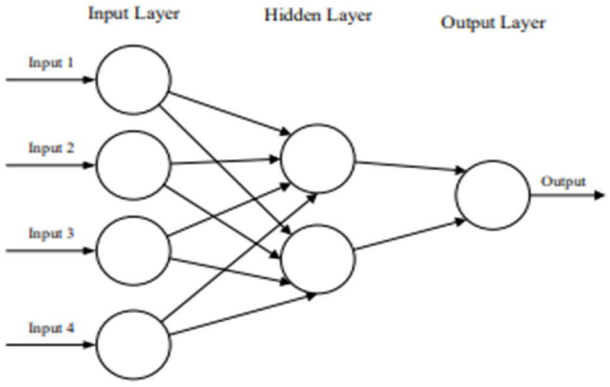


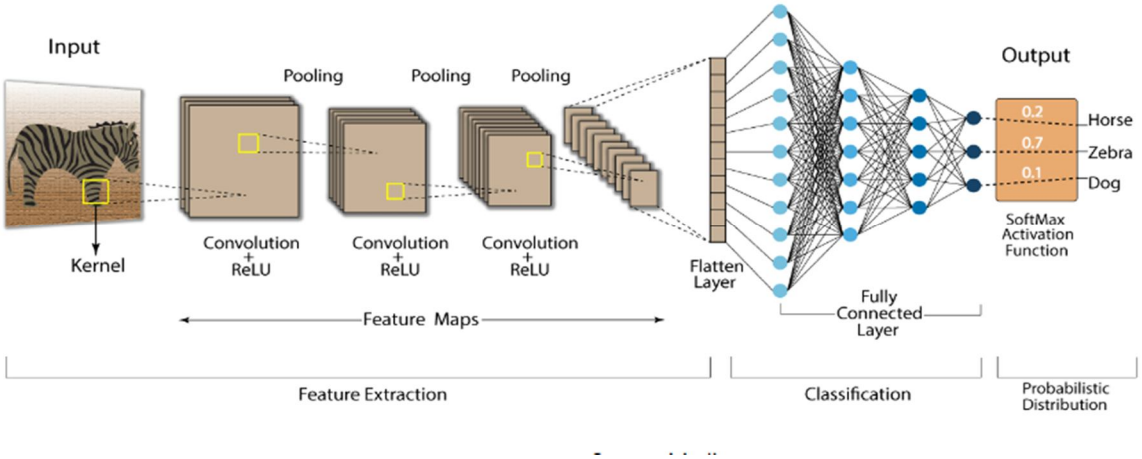
Figure.1 Feedforward Neural Network

The main advantage of a Convolution Neural Network (CNN) is reduction in parameters over the classic ANN. In classic image classification, the edges will be detected first, followed by the simpler layers in the second layer, higher level features such as faces in the third layer [8].

A. *Convolution Neural Networks*

Convolution Neural Networks also known as CNNs or ConvNets are inspired the organization of animal visual cortex. CNNs have an input layer, an output layer, many number of hidden layers, and millions of parameters which helps in learning complicated objects and patterns. The convolution layers consist of input vectors such as images, filters such as feature detectors, and output vectors such as feature maps. The image becomes a feature map, also known as activation map after passing through a convolution layer [9].

Feature Map = Input Image X Feature Detector



B. *Residual Networks*

Residual Networks (ResNets) were introduced to ease the training of large networks with substantially deep layers. The introduction of ResNets happened due to vanishing/exploding gradient problem [10]. With the network depth increasing, accuracy gets saturated and then degrades rapidly. It is also observed that such degradation is not due to overfitting. This effect can be seen in Figure 3. This degradation represents that all systems are not easy to optimize.

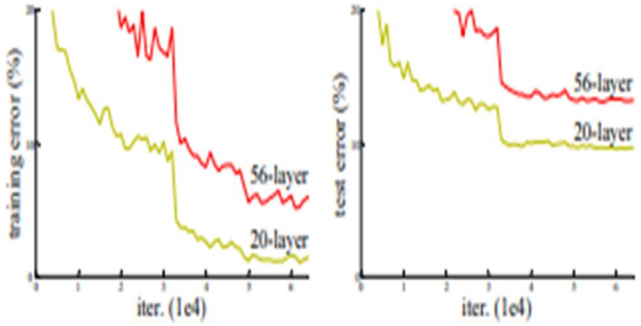


Figure 3. Training error(left) and Test error(right) for a 20-layer and 54-layer deep net on CIFAR-10.

Instead of hoping few stacked layers directly we use desired underlying map to fit residual mapping. The identity networks are used when the input and output layers are of the same dimensions [11].

At top of the module, we accept an input to the module which is nothing but the previous layer in the module. The right branch is the linear shortcut which connects the input to an addition operation at the bottom of the model. Then on the left branch of the residual module, we apply a series of convolutions, activations, batch normalizations. But with ResNets we add the original input to the output of CONV, RELU, and BN layers [12]. We call this addition an identity mapping since the input is added to output of series of operations. It is also the way the term residual is used the residual input is added to the output of series of operations [14]. The connection between input and addition node is called the shortcut. While traditional neural networks learn in the way y = f(x), a residual layer learns in the form of f(x) + id(x) = f(x) + x where id(x) is the identity function. Since the input is included in every residual layer, it turns out the network can learn faster and with larger learning rates [13]. See Figure 4.

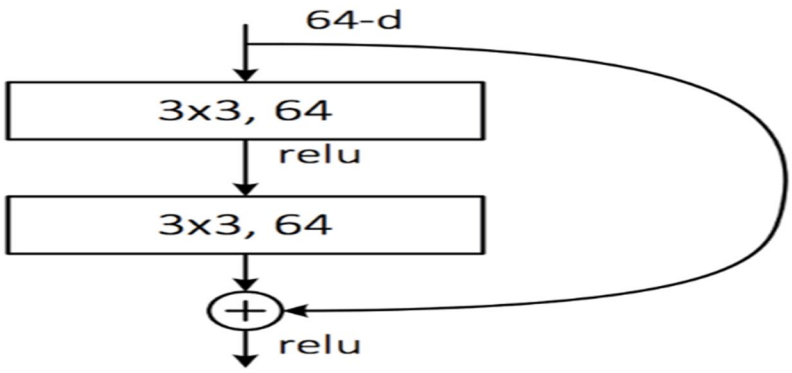


Figure 4. ResNet based on residual module.

III. IMPLEMENTATION DETAILS

In order to fine-tune ResNet with Keras and TensorFlow, we need to load ResNet from disk using the pretrained ImageNet weights but leaving off the fully-connected layer head. The final layer in the ResNet architecture is an activation layer that is 7x7x2048. We construct a new, freshly initialized layer head by accepting the base model output and then apply 7x7 average pooling followed by connected layers. Now if we take a look at model summary, we can conclude that we have successfully added a new fully-connected layer head to the ResNet. The training time is only about 240 minutes. It is measured upon the servers of Google Collaboratory.

IV. RESULTS AND DATA ANALYSIS

Most notable imports include the ResNet50 CNN architecture and Keras layers for building the head of our model for fine tuning. Settings for the entire script are hosted in the config. Additionally, we use the ImageDataGenerator class for data augmentation and scikit-learn's classification report to print stats to the terminal. We also need matplotlib for plotting and paths which assist in finding image files on disk. Data Augmentation allows for training time mutations of our images including random rotations, zooms, shifts, shears, flips, and mean subtraction. The model achieved a 97% accuracy on our normal clothes vs camouflage clothes detector as shown in Figure 6.

```
[16] # make predictions on the data
     print("[INFO] evaluating network...")
     testGen.reset()
     predIdxs = model.predict(testGen,
       steps=(totalTest // config.BS) + 1)

     # for each image in the testing set we need to find the index of the
     # label with corresponding largest predicted probability
     predIdxs = np.argmax(predIdxs, axis=1)

     # show a nicely formatted classification report
     print(classification_report(testGen.classes, predIdxs,
       target_names=testGen.class_indices.keys()))

     # serialize the model to disk
     print("[INFO] saving model...")
     model.save(config.MODEL_PATH, save_format="h5")

     [INFO] evaluating network...
                         precision    recall  f1-score   support

     camouflage_clothes       0.98      0.98      0.98      1968
         normal_clothes       0.98      0.98      0.98      2007

               accuracy                           0.98      3975
              macro avg       0.98      0.98      0.98      3975
           weighted avg       0.98      0.98      0.98      3975

     [INFO] saving model...
```

Figure 5: Metrics of the model

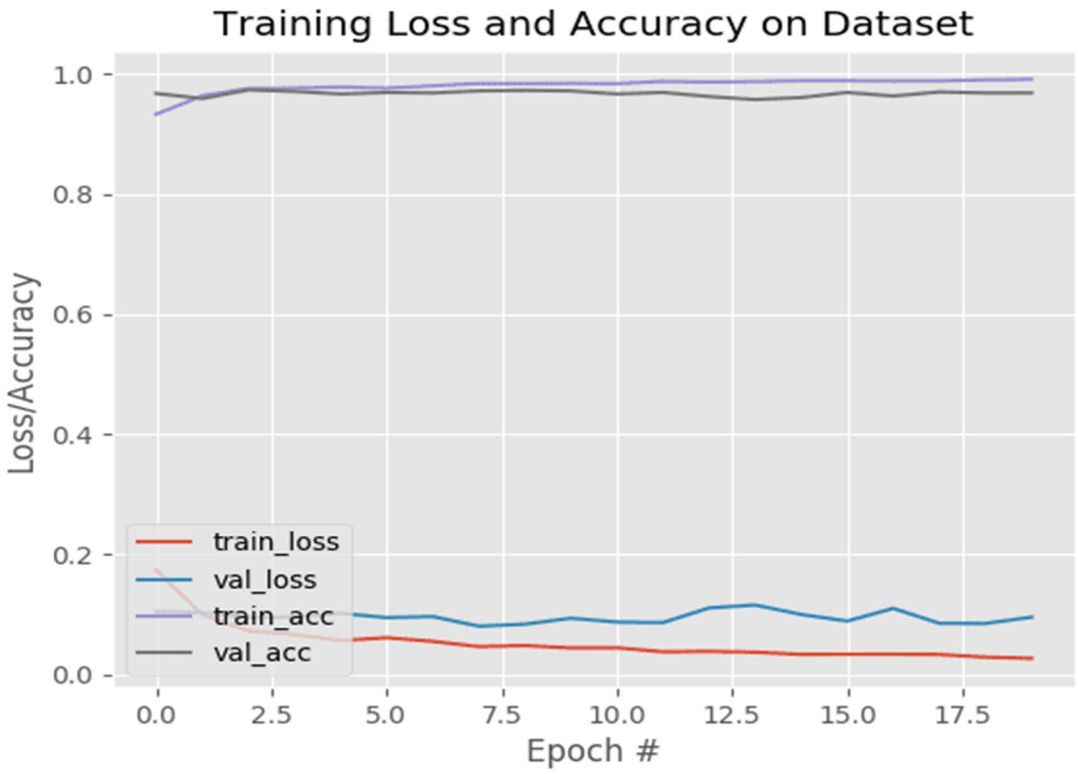
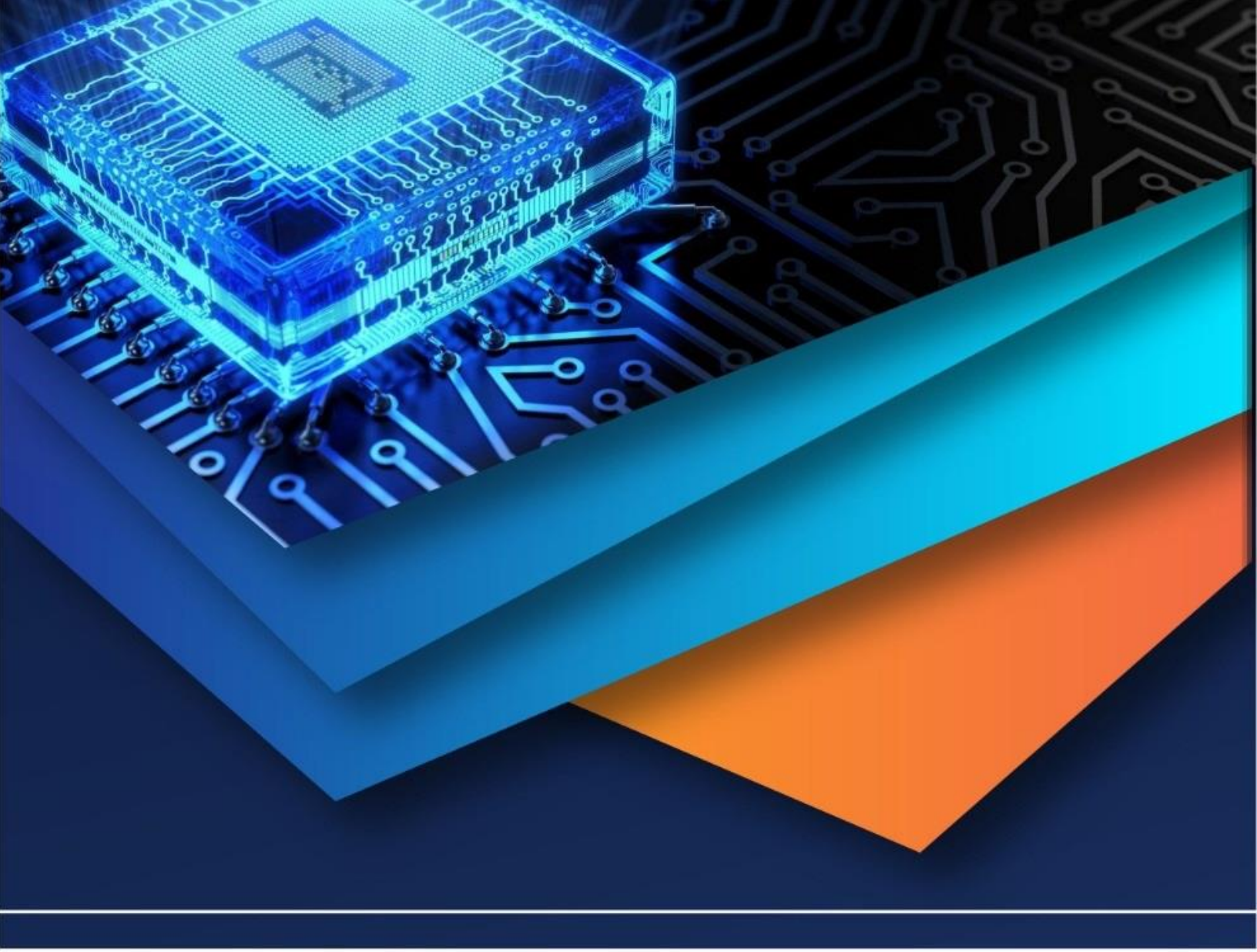


Figure 6: Training plot of our accuracy/loss curves when fine-tuning ResNet on a camouflage deep learning dataset using Keras and TensorFlow.

Our training loss decreases at a much sharper rate than our validation loss, it appears that validation loss maybe rising towards the end of training indicating that the model maybe overfitting.

REFERENCES

[1] Innes C Cuthill, Martin Stevens, Jenna Sheppard, Tracey Maddocks, C Alejandro Parraga, and Tom S Troscianko. Disruptive coloration and background pattern matching. Nature, 434(7029):72, 2005.
[2] D. -P. Fan, G. -P. Ji, G. Sun, M. -M. Cheng, J. Shen and L. Shao, "Camouflaged Object Detection," 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2020, pp. 2774-2784, doi: 10.1109/CVPR42600.2020.00285.
[3] Gerald Handerson Thayer and Abbott Handerson Thayer. Concealing-coloration in the Animal Kingdom: An Exposition of the Laws of Disguise Through Color and Pattern: Being a Summary of Abbott H. Thayer's Discoveries. Macmillan Company, 1909.
[4] Ming-Ming Cheng, Yun Liu, Wen-Yan Lin, Ziming Zhang, Paul L Rosin, and Philip HS Torr. Bing: Binarized normed gradients for objectness estimation at 300fps. Computational Visual Media, 5(1):3–20, 2019.
[5] S. Albawi, T. A. Mohammed and S. Al-Zawi, "Understanding of a convolutional neural network," 2017 International Conference on Engineering and Technology (ICET), 2017, pp. 1-6, doi: 10.1109/ICEngTechnol.2017.8308186.
[6] Y. Guo, Y. Liu, A. Oerlemans, S. Wu, and M. S. Lew, "Author's Accepted Manuscript Deep learning for visual understanding: A review To appear in: Neurocomputing," 2015.
[7] N. Kwak, "Introduction to Convolutional Neural Networks (CNNs)," 2016.
[8] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, "Going Deeper with Convolutions," 2014.
[9] K. Teilo, "An Introduction to Convolutional Neural Networks," no. NOVEMBER 2015, pp. 0–11, 2016.
[10] S. Hochreiter and J. Schmidhuber. Long short-term memory. Neural computation, 9(8):1735–1780, 1997.
[11] K. He, X. Zhang, S. Ren and J. Sun, "Deep Residual Learning for Image Recognition," 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2016, pp. 770-778, doi: 10.1109/CVPR.2016.90.
[12] R. K. Srivastava, K. Greff, and J. Schmidhuber. Training very deep networks. 1507.06228, 2015.
[13] T. Vatanen, T. Raiko, H. Valpola, and Y. LeCun. Pushing stochastic gradient towards second-order methods–backpropagation learning with transformations in nonlinearities. In Neural Information Processing, 2013.
[14] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. In ICLR, 2015.



10.22214/IJRASET

45.98

IMPACT FACTOR: 7.129

IMPACT FACTOR: 7.429

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY