A MUTUAL BOOTSTRAPPING MODEL FOR AUTOMATED SKIN LESION SEGMENTATION AND CLASSIFICATION

Report submitted to the SASTRA Deemed to be University As the requirement for the course

BCSCCS708: **MINI PROJECT**

Submitted by

ADITYA GATTU
(Reg. No.:121003008, CSE)

UPPALAPATI SAKETH
(Reg. No.: 121003295, CSE)

VAGICHERLA SAI NIKHIL
(Reg. No.: 121003296, CSE)

December 2020

SCHOOL OF COMPUTING

THANJAVUR, TAMIL NADU, INDIA – 613 401

SCHOOL OF COMPUTING THANJAVUR-613 401

<u>Bonafide Certificate</u>

This is to certify that the report titled "**A MUTUAL BOOTSTRAPPING MODEL FOR AUTOMATED SKIN LESION SEGMENTATION AND CLASSIFICATION**" submitted as a requirement for the course, BCSCCS708: **MINI PROJECT** for B.Tech. is a bonafide record of the work done by **Mr. ADITYA GATTU (Reg. No: 121003008, CSE), Mr. UPPALAPATI SAKETH (Reg. No: 121003295, CSE),**

Mr. VAGICHERLA SAI NIKHIL (Reg. No: 121003296, CSE) during the academic year 2020-21, in the School of Computing, under my supervision.

Signature of the Project Supervisor:

Name with Affiliation :

Date :

Mini Project *Viva voc*e held on

Examiner 1 Examiner 2

ACKNOWLEDGEMENT

We would like to express our sincere gratitude to Dr. S. Vaidhyasubramaniam, Vice Chancellor for his encouragement during the span of our academic life at SASTRA Deemed to be University.

We were extremely grateful to Dr. A Umamakeswari, Dean, School of Computing. and Dr. V.S. Shankar Sriram, Associate Dean, Department of Computer Science and Engineering for their academic support and a constant motivation for the entire academic year in School of Computing.

We express our sincere thanks to our project guide Dr. Premalatha.J , AP-II, IT, School of Computing, who guided us from the inception till the successful accomplishment of this project. Working on this project helped us to perceive the immense experience.

We also like to acknowledge the open resources in the internet and the Swayam course (Deep learning for computer Vision) which helped us for completion of our project.

Finally. We thank God Almighty, our Parents who helped us in successfully completing this project.

List of Figures

List of Tables

ABBREVIATIONS

AUC	Area Under Curve
BN	Batch Normalization
CAD	Computer Aided Diagnosis
CAM	Class Activation Map
CN	Classification Network
CNN	Convolutional Neural Network
FC	Fully Connected layer
GAP	Global Average Pooling
MB - DCNN	Mutual Bootstrapping Deep Convolutional Neural Network
SN	Segmentation Network

NOTATIONS

Xn	Representation of the images for segmentation
Yn	Image level label for segmentation
Xm	Representation of the images for classification
Ym	Image level label for classification
N1	Images in segmentation training set
N2	Images in classification training set
In	Dataset
C	Number of classes
λ	Weighting Factor

ABSTRACT

The project mainly focuses on diagnosis of Skin Cancer. It commonly occurs due to the abnormal growth of the cells/tissues. One of the most common skin cancers today is Melanoma.. Melanoma is the most common type of cancer that is prevalent nowadays. It is difficult to diagnose the melanoma at the initial stage because of its similarity to a normal mole.Hence, an intelligent model with deep learning models are proposed for melanoma diagnosis. This project work uses dermoscopy images from PH2 and ISIC image datasets. The common challenge that is encountered in these image datasets is localizing and enhancing the images.

In this same way, we utilize the power of bootstrapping in a mutual way by using the (MB-CNN) model for synchronous segmentation of skin lesions followed by classification. It tends to be accomplished through three phases as coarse SN, secondly mask CN, and finally enhancement of segmentation network enhanced SN. Firstly, the coarse-SN creates a lesion cover for the mask-CN which precisely finds and orders skin lesions. Secondly, the lesion restriction delivered in the mask CN is taken care of into the improved SN, focusing on exchanging confinement information to the improved SN. Segmentation network and classification network move information between one another and encourage themselves in bootstrapping. Finally we assessed the DCNN model on the HAM-ISIC 2017 dataset and accomplished AUC of 95.74% on an average.

Keywords:

Cancer Diagnosis, Segmentation, Feature Mask generation, Classification, DeepLabv3, Mutual Bootstrapping model.

Table of Contents

BASE PAPER DETAILS

Authors : Yutong Xie†, Jianpeng Zhang†, Yong Xia, and Chunhua Shen.

Title :A Mutual Bootstrapping Model for Automated Skin Lesion Segmentation and Classification

DOI:10.1109/TMI.2020.2972964

Date of Publication:10 February 2020

Impact Factor:6.685

Journal :IEEE TRANSACTIONS ON MEDICAL IMAGING

BASE PAPER SUMMARY

Cancer related to skin is one of the most widely recognized malignancies influencing humankind around the world. Dermoscopic pictures are concentrated by dermatologists to recognize the malignancy. Yet, this cycle is helped out completely through visual review which requires a serious level of expertise and fixation and is a tedious cycle.

Computer-aided diagnosis (CAD) comes into help in helping the dermatologists in bypassing these issues as well as improving the exactness, proficiency and objectivity of the finding.

There are two huge assignments to build up a CAD arrangement of skin disease: skin sore division and characterization. The division cycle is utilized to recognize the areas and limits of sores, though the characterization task is to recognize the sorts among them (for example melanoma, nevus, seborrheic keratosis, and so on)

There are mainly three tasks that are challenging, those are:

(1) The detection of lesion boundaries is challenging as there is minimal contrast encountered in every lesion and the corresponding encompassing tissues of skin.

(2) Visual similarities may be shared among the inter-type skin lesions while visual similarities can be seen among the intra-type lesions.

(3)The components like hair, outlines, in blood vessels, and air bubbles may affect the visual impact of skin lesions as the image can be corrupted if it has any of the components.

The proposed model is mutual bootstrapping DCNN model for simultaneously do segmentation of skin lession as well as classification , we believe, a very efficient method. The model comprises of the coarse SN, a mask CN, and finally the enhanced segmentation network.

The principle focal point of Skin lesion classification strategies is on removing carefully assembled highlights, including the shading, outskirt anomaly, surface, and imbalance descriptors of injuries. It likewise centers around utilizing at least one of these highlights to prepare a classifier, for example, the K-closest neighbor.

For training the set consists of N1 images can be represented by IN = (Xn,Yn) N1 , here each image Xn is explained on pixel by pixel premise and every pixel has a place with either the skin lesion (i.e. yni=1) or background (i.e. yni=0).

For classification the training set with N2 (N2 > N1) pictures be signified by IM = (Xm,Ym) N2 , in which each picture Xm is clarified with a picture level mark ym $\in$ {l1, ..., lC }, and C is the quantity of classes.

Presently, coarse-SN is prepared on the IN informational collection for lesion division. I kThen, to improve its proficiency in lesion arrangement, we combine the pictures in IM with the relating sore covers created by coarse-SN and this is taken as the maskCN input. At long last, we utilize an improved layer E layer to consolidate the highlights of pictures IN recovered by the upgraded SN encoder and the comparing maskCN refined lesion position maps, and afterward feed them to the improved SN decoder for more definite division of injuries.

Coarse-SN:

First, coarse-SN is equipped to approximately segment skin lesions using the IN dataset. The masks of coarse SN gives us mask CN with a prior knowledge on lesion sites, which will thus boost the capacity of mask-CN to localize and differentiate.

We detach the last convolutional layer to fit the Deeplabv3+ network in order for the completion of skin lesion segmentation task, next we add a new layer which gives us an output through a single channel to obtain prediction. The new layer's weights are initialized at random and the sigmoid activation function is used.

Mask-CN:

In order to increase the localization of the skin lesion and discriminating capacity of mask classification network ,that is prepared on IM dataset, in which every picture has just a picture based class level mark, we utilize the masks of coarse lesions made by coarse-SN. The skin sore region generally incorporates just a little part of the skin image, and the most zones of the picture are characteristic skin tissues containing items, for example, hair, outlines, veins, and air bubbles that can strife with the characterization of the injuries.

As a contribution to maskCN, every classification training picture Xm and its comparing coarse lesion mask are linked. The created features are taken care of to another layer which is fully connected using C neurons, trailed by activation mechanism and the type is softmax, in the wake of playing out the global average pooling (GAP). The FC layer's loads are introduced arbitrarily. The information weights of the fourth channel (for example coarse veil) are introduced by finding the normal weights of the channels. We streamline the mask CN by limiting the cross entropy loss.

Enhanced-SN:

We provide this result into the qualified mask CN for each segmentation training image Xn and use this result of the last convolutional layer to generate the class based localization maps Mn through the use of the CAM . Specifically, by using the class-specific weights of the output layer, we initially gauge the component maps created by the last layer of mask CN, and afterward summarize weighted element maps over all the paths to produce class explicit limitation maps Mn.

For coarse-SN, the encoder and decoder share similar design and boundaries. The E layer initially links element maps furnished using encoder for Mn to coordinate the refined earlier data about sore situations in Mn into the division stage and afterward utilizes a 1 x 1 convolution layer and then cluster standardization using BN and also ReLU initiation system for data combination, for example intertwining the elevated level picture highlights and injury restriction data.

Hybrid loss for coarse-SN and enhanced-SN:

We suggest the accompanying hybrid-loss feature to improve both enhanced SN and coarse SN.

$$L_{hybrid} = L_{dice} + \lambda L_{rank} \tag{1}$$

where Lrank is a rank loss, Ldice is the loss of the dice, and λ is an arbitrary weighting factor that governs the contribution of Lrank's.

The Dice-loss decides the level of arrangement, appeared as follows, between the expectation and the actual truth.

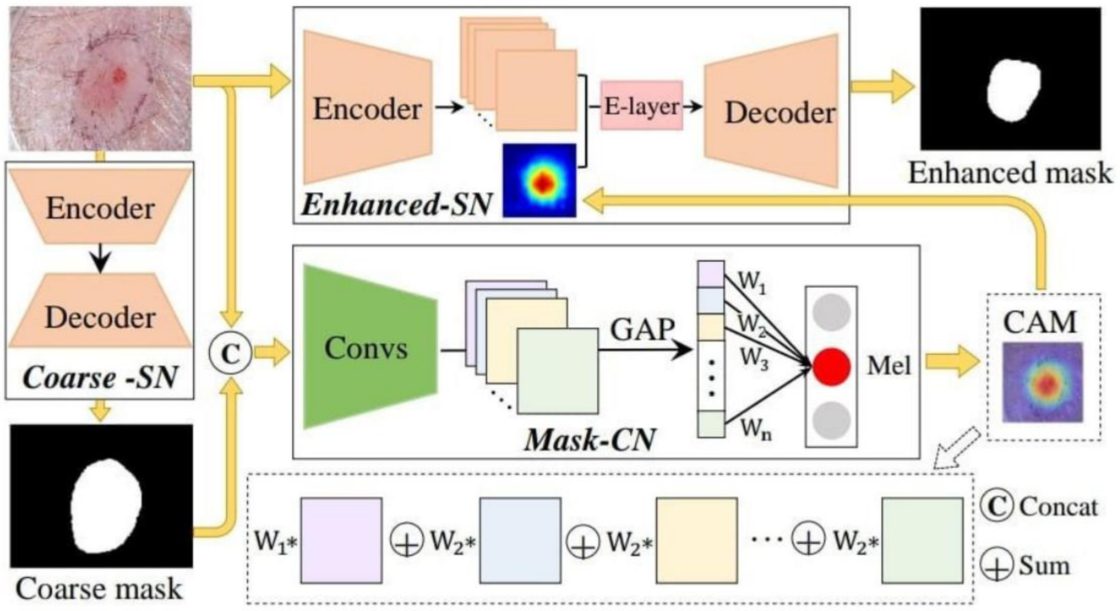
$$\mathcal{L}_{dice} = 1 - \frac{2\sum_{i=1}^{V} p_i y_i}{\sum_{i=1}^{V}(p_i + y_i) + \varepsilon} \quad (2)$$

Here, Pi represents normal probability of ith pixel having a place with the sore, yi speaks to the ground reality characteristic of the pixel in ith position, and ε represents smooth part where V indicates the quantity of pixels.

The figured rank-error is given by the equation

$$L_{rank}(\mathbf{X}_n, \mathbf{Y}_n) = \frac{1}{K^2}\sum_{i=1}^{K}\sum_{j=1}^{K}\max\{0, \mathbf{H}_{ni}^{0}(\mathbf{X}_n, \mathbf{Y}_n) - \mathbf{H}_{nj}^{1}(\mathbf{X}_n, \mathbf{Y}_n) + margin\} \quad (3)$$

Architecture that is being used



MERITS

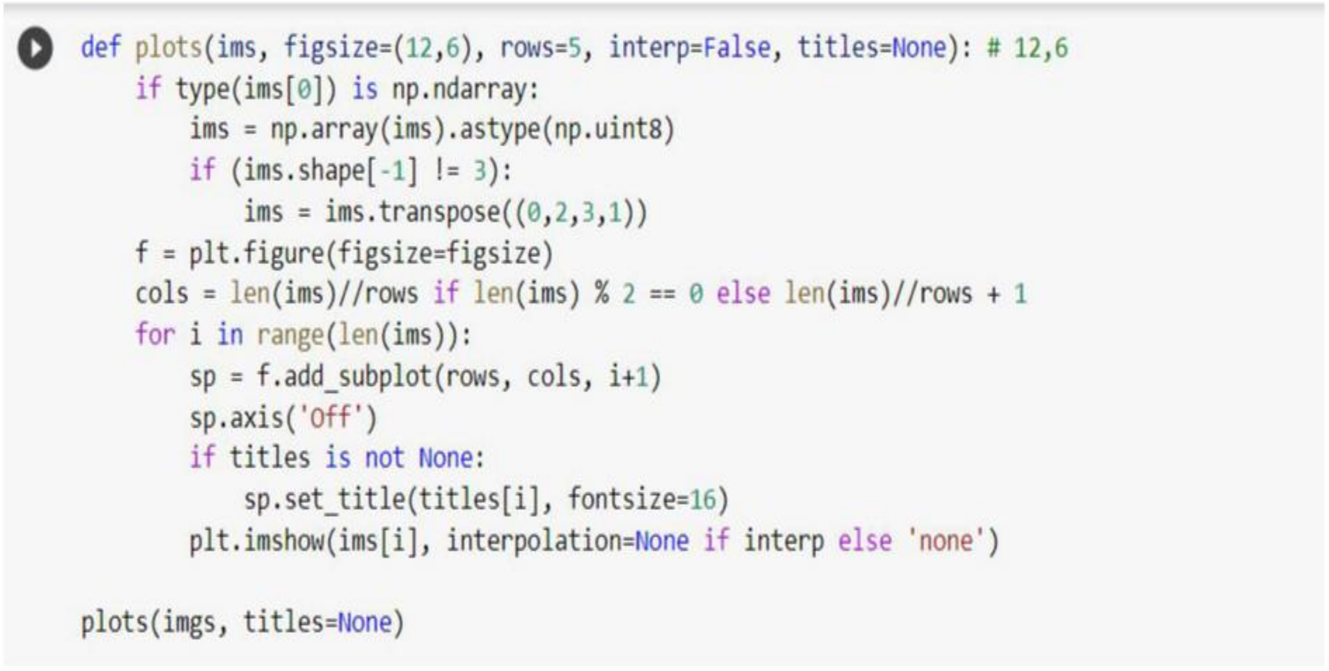
- Mutual Bootstrapping model uses the coarse mask produced through segmentation to mask classification networks to classify the lesions accurately.
- This transfer of obtained images to enhanced segmentation helps to reduce the inaccurate results and to obtain correct localization of lesions.
- Decoders in the model combine the features that are extracted from different atrous convolutions.So,that we can’t miss the miniature important feature from the image.
- Our model uses the separable convolution helps in reducing the number of computations.
- Deeplab uses the atrous convolution which helps in looking at broder space of images.i.e ,features from all over the images are convolved.
- For the segmentation we need less number of pixel level dense annotation.To achieve this we will compare it with the supervised based segmentation without having the feature maps.
- Segmentation helps in pictorial representation of images to interpret images better by humans.
- In the classification the discriminative features are extracted from mask CN.Although,the classification helps in finding regions of interest.
- By this mutual compensation of mask CN from coarse SN helps in Mutual Bootstrapping model.

DEMERITS

- CNN’s don’t remember the exact position and the orientations of the image.
- CNNS are more prone to the adversarial attacks and achieving the robustness of the model is a very difficult thing.
- The images may vary from region to region.So,that achieving same accuracy for different datasets is also uncertain.
- Segmentation is difficult for applying trivial images.
- Segmentation is difficult because there is lack of context in the images.Thresholding may not work in all the cases.

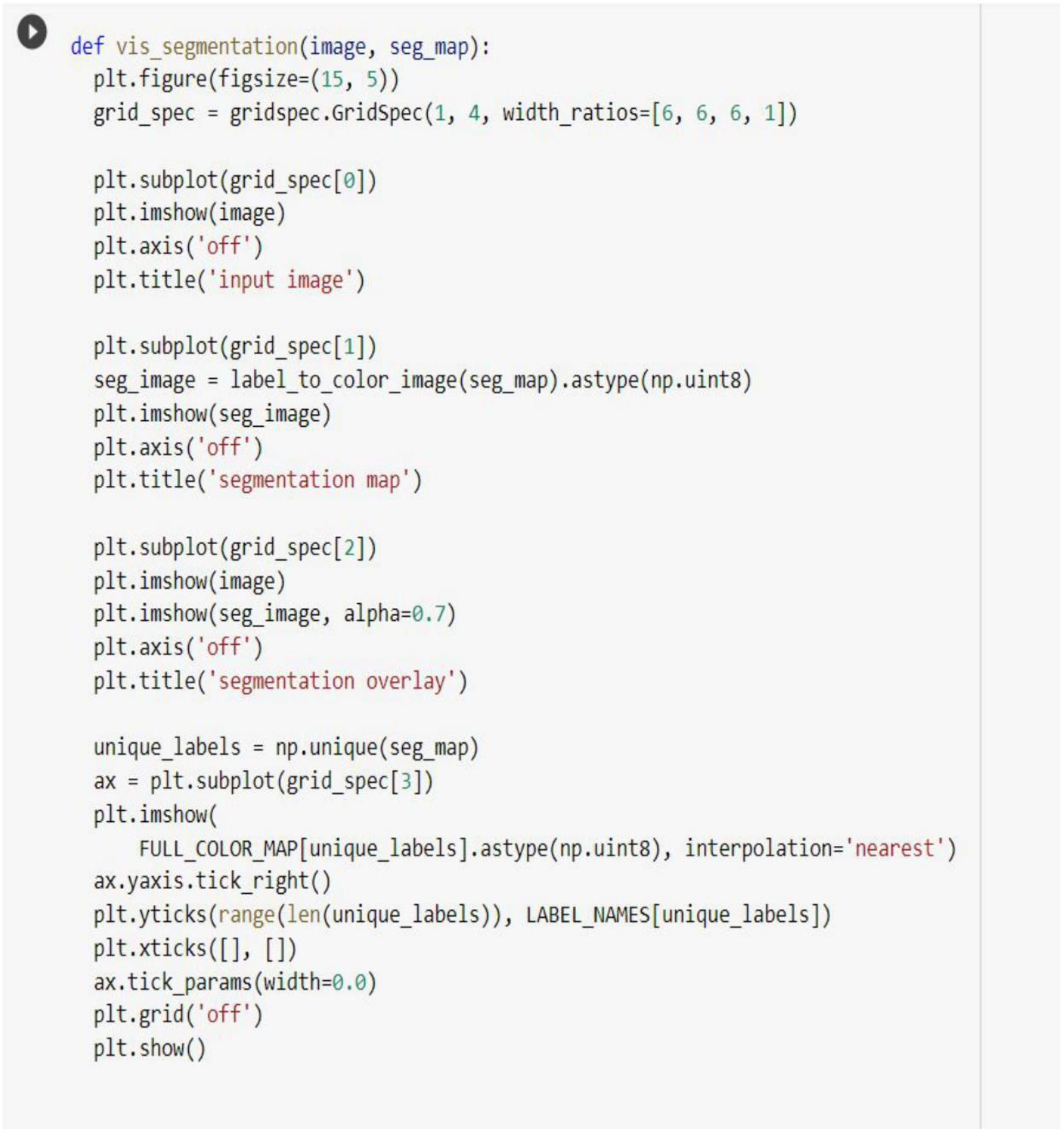
Source Code

Visualize the images :



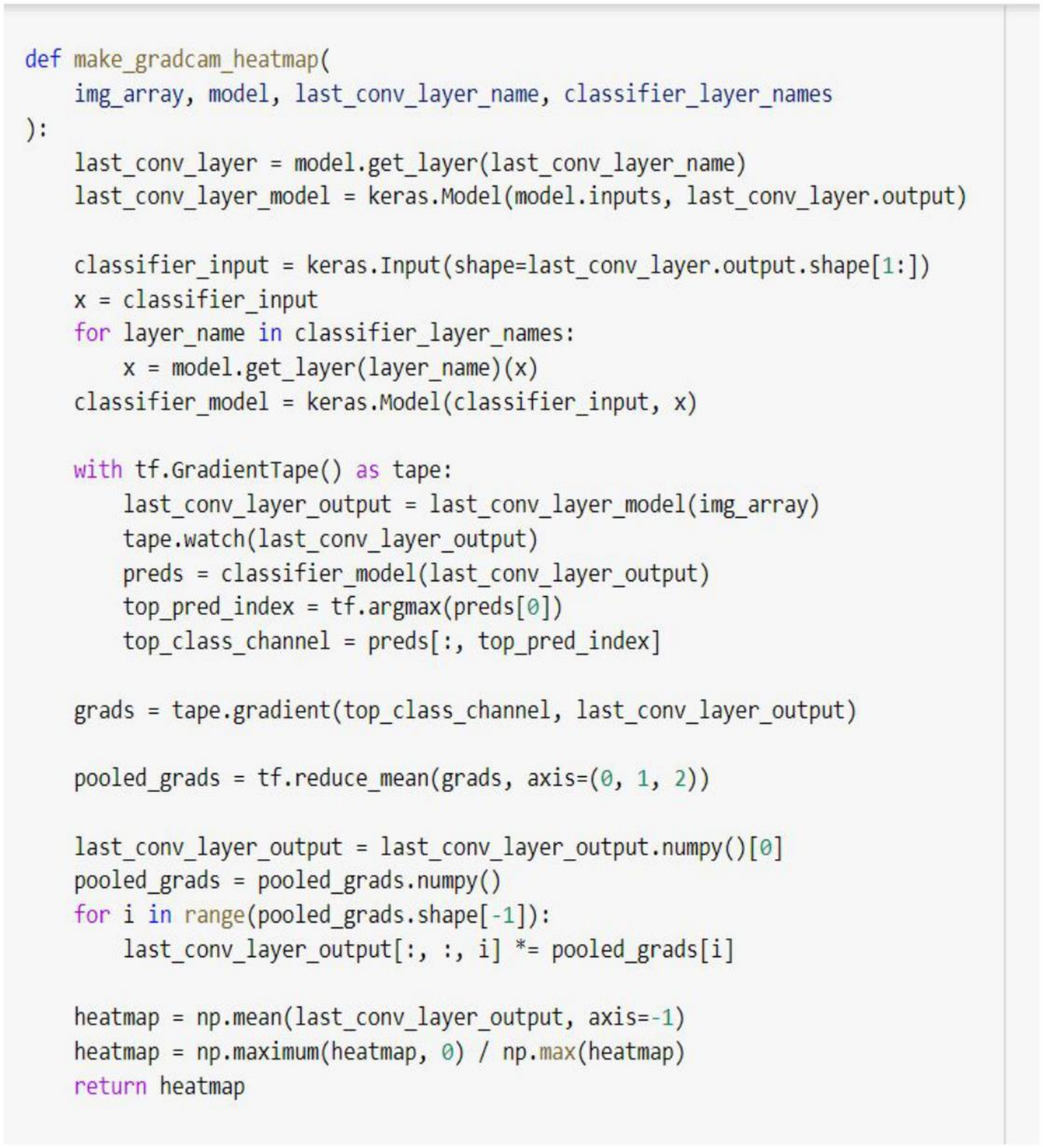
```python
def plots(ims, figsize=(12,6), rows=5, interp=False, titles=None): # 12,6
    if type(ims[0]) is np.ndarray:
        ims = np.array(ims).astype(np.uint8)
        if (ims.shape[-1] != 3):
            ims = ims.transpose((0,2,3,1))
    f = plt.figure(figsize=figsize)
    cols = len(ims)//rows if len(ims) % 2 == 0 else len(ims)//rows + 1
    for i in range(len(ims)):
        sp = f.add_subplot(rows, cols, i+1)
        sp.axis('Off')
        if titles is not None:
            sp.set_title(titles[i], fontsize=16)
        plt.imshow(ims[i], interpolation=None if interp else 'none')

plots(imgs, titles=None)
```

Segmentation :

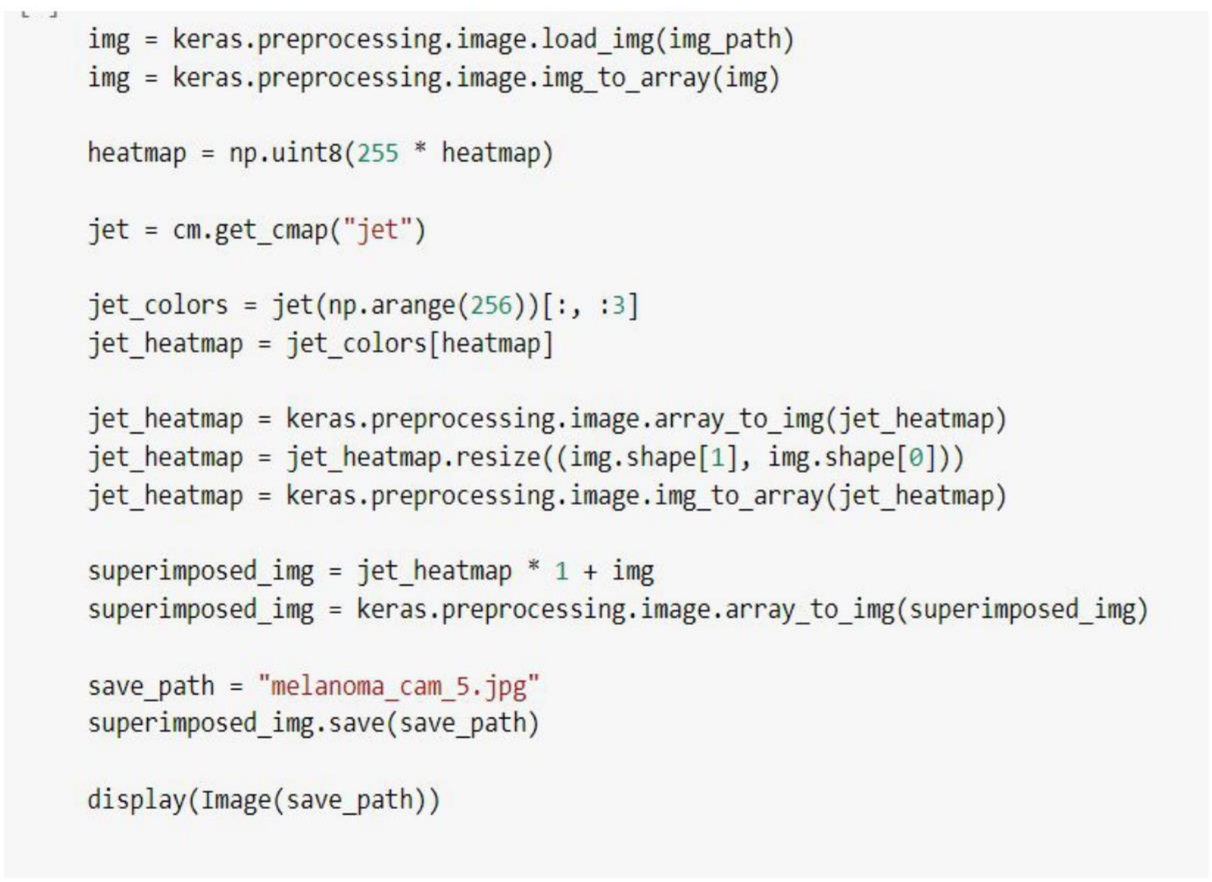


```python
def vis_segmentation(image, seg_map):
  plt.figure(figsize=(15, 5))
  grid_spec = gridspec.GridSpec(1, 4, width_ratios=[6, 6, 6, 1])

  plt.subplot(grid_spec[0])
  plt.imshow(image)
  plt.axis('off')
  plt.title('input image')

  plt.subplot(grid_spec[1])
  seg_image = label_to_color_image(seg_map).astype(np.uint8)
  plt.imshow(seg_image)
  plt.axis('off')
  plt.title('segmentation map')

  plt.subplot(grid_spec[2])
  plt.imshow(image)
  plt.imshow(seg_image, alpha=0.7)
  plt.axis('off')
  plt.title('segmentation overlay')

  unique_labels = np.unique(seg_map)
  ax = plt.subplot(grid_spec[3])
  plt.imshow(
      FULL_COLOR_MAP[unique_labels].astype(np.uint8), interpolation='nearest')
  ax.yaxis.tick_right()
  plt.yticks(range(len(unique_labels)), LABEL_NAMES[unique_labels])
  plt.xticks([], [])
  ax.tick_params(width=0.0)
  plt.grid('off')
  plt.show()
```

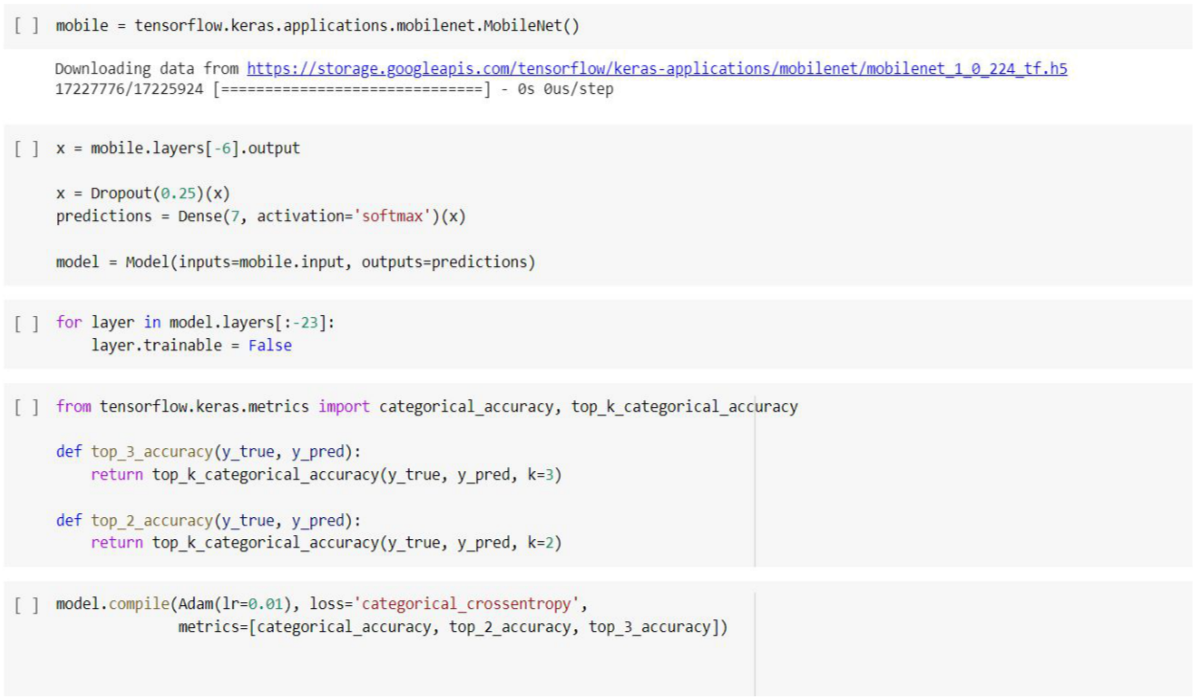
CAM Model Generation:



```python
def make_gradcam_heatmap(
    img_array, model, last_conv_layer_name, classifier_layer_names
):
    last_conv_layer = model.get_layer(last_conv_layer_name)
    last_conv_layer_model = keras.Model(model.inputs, last_conv_layer.output)

    classifier_input = keras.Input(shape=last_conv_layer.output.shape[1:])
    x = classifier_input
    for layer_name in classifier_layer_names:
        x = model.get_layer(layer_name)(x)
    classifier_model = keras.Model(classifier_input, x)

    with tf.GradientTape() as tape:
        last_conv_layer_output = last_conv_layer_model(img_array)
        tape.watch(last_conv_layer_output)
        preds = classifier_model(last_conv_layer_output)
        top_pred_index = tf.argmax(preds[0])
        top_class_channel = preds[:, top_pred_index]

    grads = tape.gradient(top_class_channel, last_conv_layer_output)

    pooled_grads = tf.reduce_mean(grads, axis=(0, 1, 2))

    last_conv_layer_output = last_conv_layer_output.numpy()[0]
    pooled_grads = pooled_grads.numpy()
    for i in range(pooled_grads.shape[-1]):
        last_conv_layer_output[:, :, i] *= pooled_grads[i]

    heatmap = np.mean(last_conv_layer_output, axis=-1)
    heatmap = np.maximum(heatmap, 0) / np.max(heatmap)
    return heatmap
```


```
img = keras.preprocessing.image.load_img(img_path)
img = keras.preprocessing.image.img_to_array(img)

heatmap = np.uint8(255 * heatmap)

jet = cm.get_cmap("jet")

jet_colors = jet(np.arange(256))[:, :3]
jet_heatmap = jet_colors[heatmap]

jet_heatmap = keras.preprocessing.image.array_to_img(jet_heatmap)
jet_heatmap = jet_heatmap.resize((img.shape[1], img.shape[0]))
jet_heatmap = keras.preprocessing.image.img_to_array(jet_heatmap)

superimposed_img = jet_heatmap * 1 + img
superimposed_img = keras.preprocessing.image.array_to_img(superimposed_img)

save_path = "melanoma_cam_5.jpg"
superimposed_img.save(save_path)

display(Image(save_path))
```

Classification:



```
[ ] mobile = tensorflow.keras.applications.mobilenet.MobileNet()

    Downloading data from https://storage.googleapis.com/tensorflow/keras-applications/mobilenet/mobilenet_1_0_224_tf.h5
    17227776/17225924 [==============================] - 0s 0us/step
```

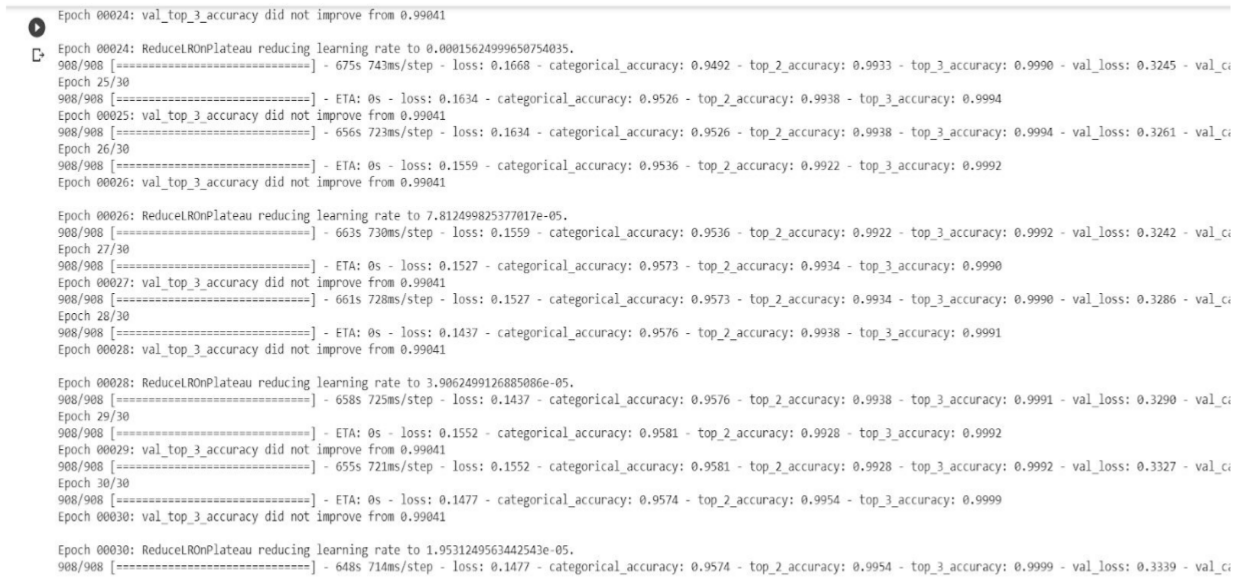
```
[ ] x = mobile.layers[-6].output

    x = Dropout(0.25)(x)
    predictions = Dense(7, activation='softmax')(x)

    model = Model(inputs=mobile.input, outputs=predictions)
```

```
[ ] for layer in model.layers[:-23]:
        layer.trainable = False
```

```
[ ] from tensorflow.keras.metrics import categorical_accuracy, top_k_categorical_accuracy

    def top_3_accuracy(y_true, y_pred):
        return top_k_categorical_accuracy(y_true, y_pred, k=3)

    def top_2_accuracy(y_true, y_pred):
        return top_k_categorical_accuracy(y_true, y_pred, k=2)
```

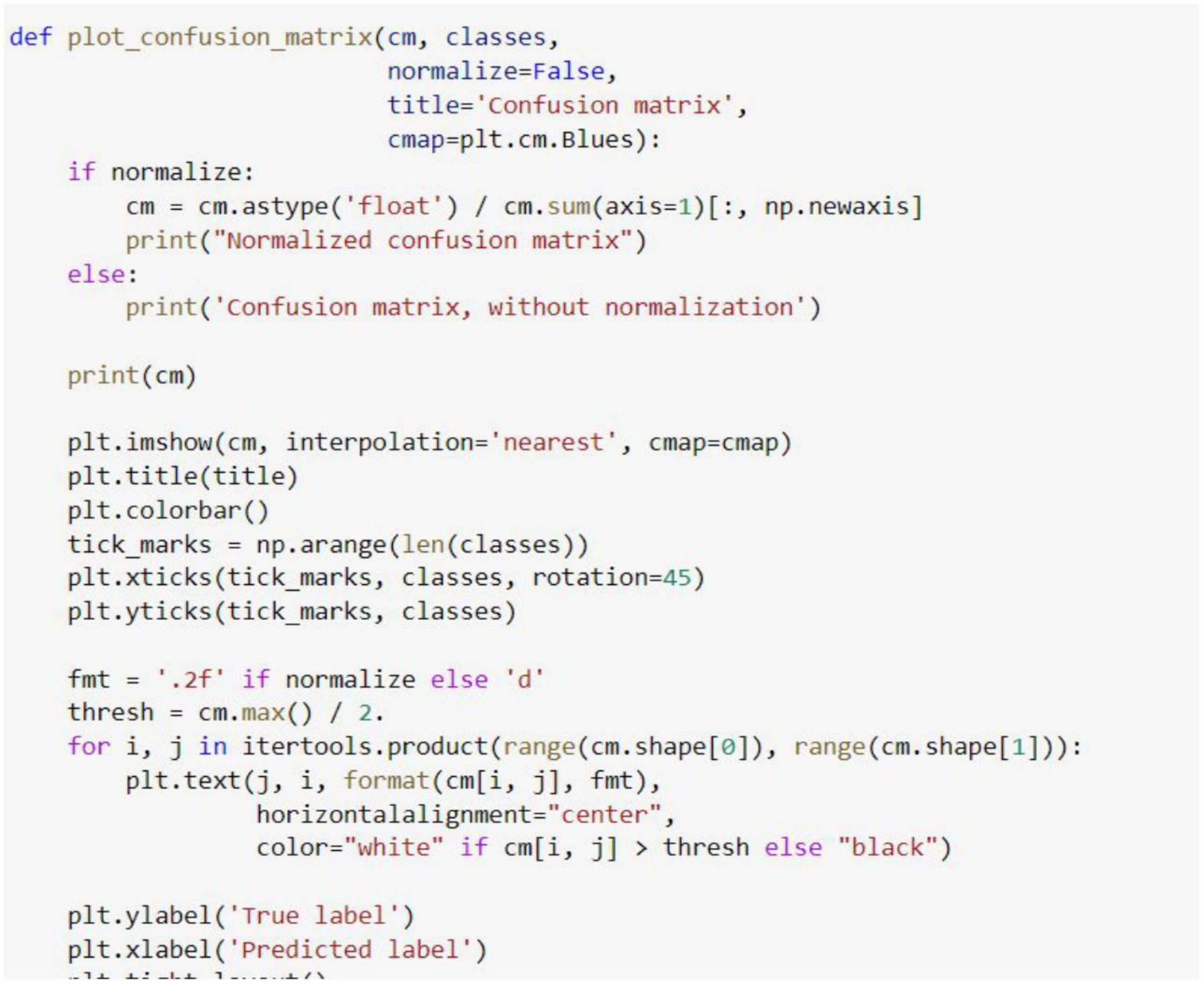
```
[ ] model.compile(Adam(lr=0.01), loss='categorical_crossentropy',
                  metrics=[categorical_accuracy, top_2_accuracy, top_3_accuracy])
```

Training Phase:



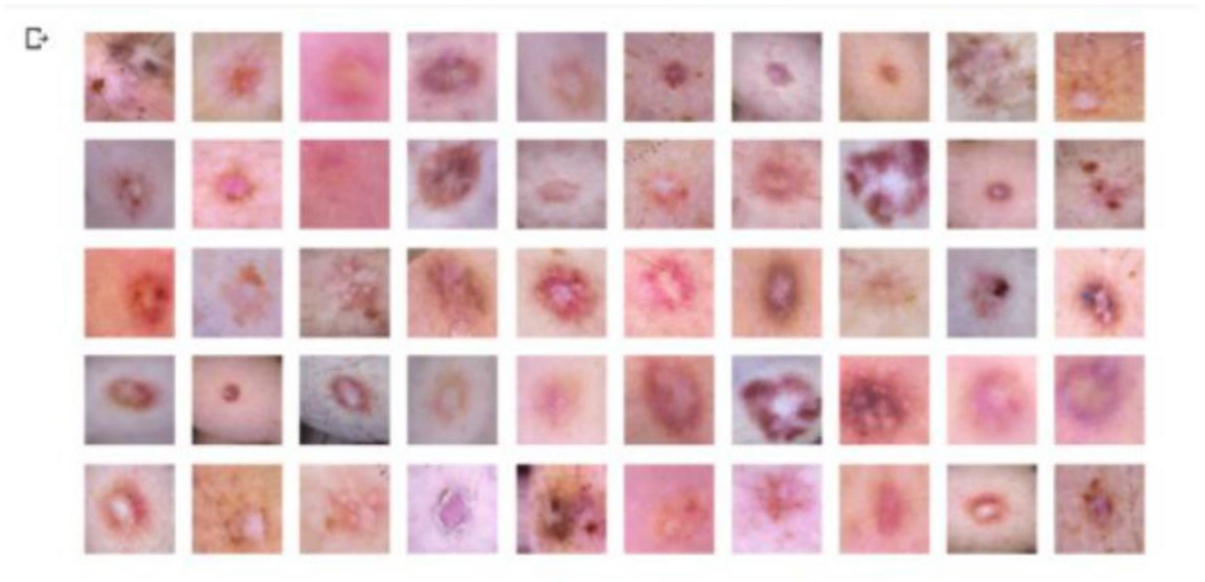
```
Epoch 00024: val_top_3_accuracy did not improve from 0.99041

Epoch 00024: ReduceLROnPlateau reducing learning rate to 0.00015624999650754035.
908/908 [==============================] - 675s 743ms/step - loss: 0.1668 - categorical_accuracy: 0.9492 - top_2_accuracy: 0.9933 - top_3_accuracy: 0.9990 - val_loss: 0.3245 - val_ca
Epoch 25/30
908/908 [==============================] - ETA: 0s - loss: 0.1634 - categorical_accuracy: 0.9526 - top_2_accuracy: 0.9938 - top_3_accuracy: 0.9994
Epoch 00025: val_top_3_accuracy did not improve from 0.99041
908/908 [==============================] - 656s 723ms/step - loss: 0.1634 - categorical_accuracy: 0.9526 - top_2_accuracy: 0.9938 - top_3_accuracy: 0.9994 - val_loss: 0.3261 - val_ca
Epoch 26/30
908/908 [==============================] - ETA: 0s - loss: 0.1559 - categorical_accuracy: 0.9536 - top_2_accuracy: 0.9922 - top_3_accuracy: 0.9992
Epoch 00026: val_top_3_accuracy did not improve from 0.99041

Epoch 00026: ReduceLROnPlateau reducing learning rate to 7.812499825377017e-05.
908/908 [==============================] - 663s 730ms/step - loss: 0.1559 - categorical_accuracy: 0.9536 - top_2_accuracy: 0.9922 - top_3_accuracy: 0.9992 - val_loss: 0.3242 - val_ca
Epoch 27/30
908/908 [==============================] - ETA: 0s - loss: 0.1527 - categorical_accuracy: 0.9573 - top_2_accuracy: 0.9934 - top_3_accuracy: 0.9990
Epoch 00027: val_top_3_accuracy did not improve from 0.99041
908/908 [==============================] - 661s 728ms/step - loss: 0.1527 - categorical_accuracy: 0.9573 - top_2_accuracy: 0.9934 - top_3_accuracy: 0.9990 - val_loss: 0.3286 - val_ca
Epoch 28/30
908/908 [==============================] - ETA: 0s - loss: 0.1437 - categorical_accuracy: 0.9576 - top_2_accuracy: 0.9938 - top_3_accuracy: 0.9991
Epoch 00028: val_top_3_accuracy did not improve from 0.99041

Epoch 00028: ReduceLROnPlateau reducing learning rate to 3.9062499126885086e-05.
908/908 [==============================] - 658s 725ms/step - loss: 0.1437 - categorical_accuracy: 0.9576 - top_2_accuracy: 0.9938 - top_3_accuracy: 0.9991 - val_loss: 0.3290 - val_ca
Epoch 29/30
908/908 [==============================] - ETA: 0s - loss: 0.1552 - categorical_accuracy: 0.9581 - top_2_accuracy: 0.9928 - top_3_accuracy: 0.9992
Epoch 00029: val_top_3_accuracy did not improve from 0.99041
908/908 [==============================] - 655s 721ms/step - loss: 0.1552 - categorical_accuracy: 0.9581 - top_2_accuracy: 0.9928 - top_3_accuracy: 0.9992 - val_loss: 0.3327 - val_ca
Epoch 30/30
908/908 [==============================] - ETA: 0s - loss: 0.1477 - categorical_accuracy: 0.9574 - top_2_accuracy: 0.9954 - top_3_accuracy: 0.9999
Epoch 00030: val_top_3_accuracy did not improve from 0.99041

Epoch 00030: ReduceLROnPlateau reducing learning rate to 1.9531249563442543e-05.
908/908 [==============================] - 648s 714ms/step - loss: 0.1477 - categorical_accuracy: 0.9574 - top_2_accuracy: 0.9954 - top_3_accuracy: 0.9999 - val_loss: 0.3339 - val_ca
```

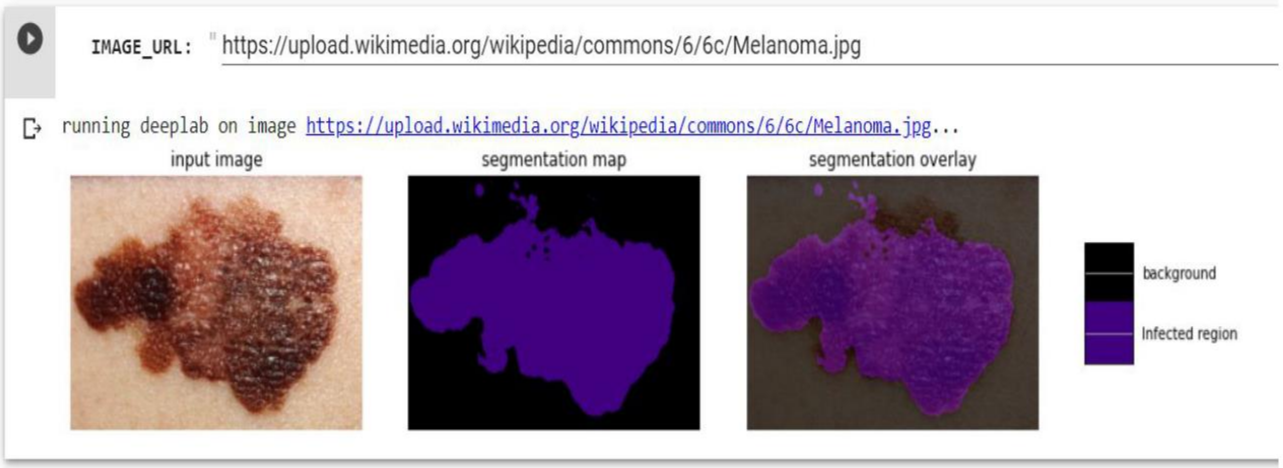
Confusion Matrix Generation:



```
def plot_confusion_matrix(cm, classes,
                          normalize=False,
                          title='Confusion matrix',
                          cmap=plt.cm.Blues):
    if normalize:
        cm = cm.astype('float') / cm.sum(axis=1)[:, np.newaxis]
        print("Normalized confusion matrix")
    else:
        print('Confusion matrix, without normalization')

    print(cm)

    plt.imshow(cm, interpolation='nearest', cmap=cmap)
    plt.title(title)
    plt.colorbar()
    tick_marks = np.arange(len(classes))
    plt.xticks(tick_marks, classes, rotation=45)
    plt.yticks(tick_marks, classes)

    fmt = '.2f' if normalize else 'd'
    thresh = cm.max() / 2.
    for i, j in itertools.product(range(cm.shape[0]), range(cm.shape[1])):
        plt.text(j, i, format(cm[i, j], fmt),
                 horizontalalignment="center",
                 color="white" if cm[i, j] > thresh else "black")

    plt.ylabel('True label')
    plt.xlabel('Predicted label')
```

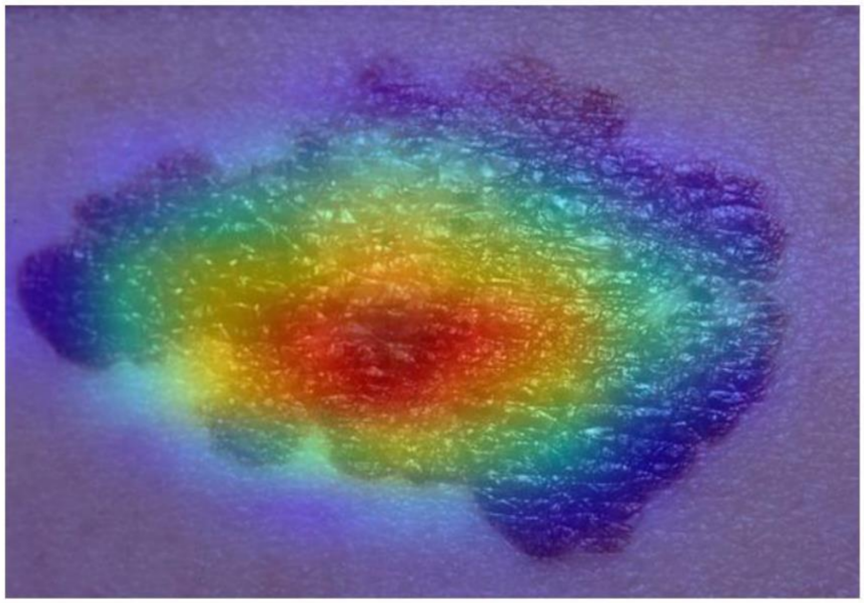
Snapshots:



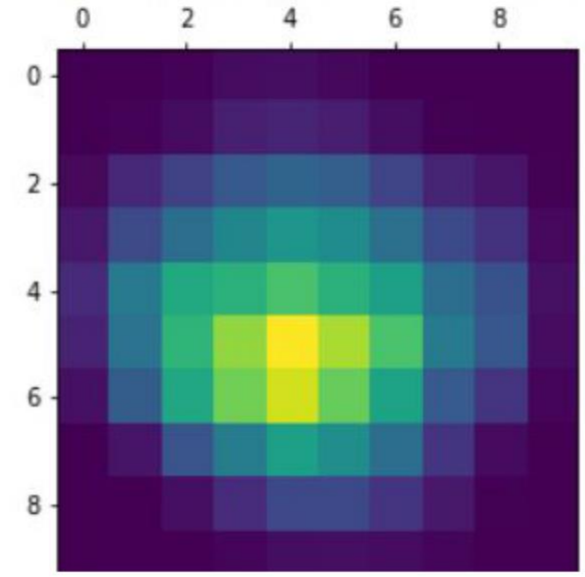


```
IMAGE_URL: "https://upload.wikimedia.org/wikipedia/commons/6/6c/Melanoma.jpg
```

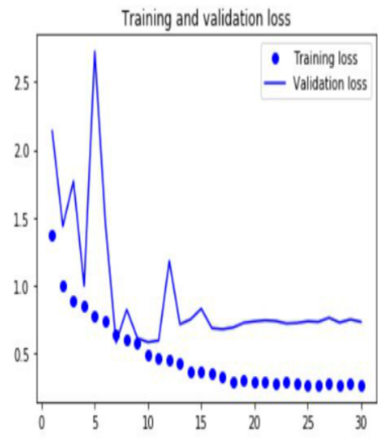
```
running deeplab on image https://upload.wikimedia.org/wikipedia/commons/6/6c/Melanoma.jpg...
```

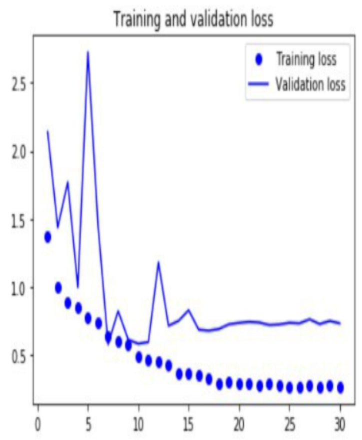


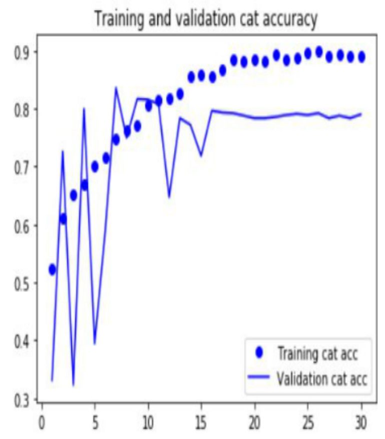
```
Predicted: [('n01776313', 'tick', 0.20322382)]
```

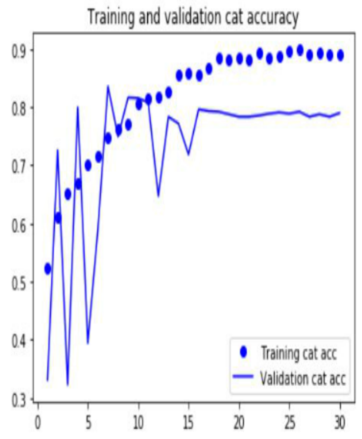


Training Vs Validation Loss :

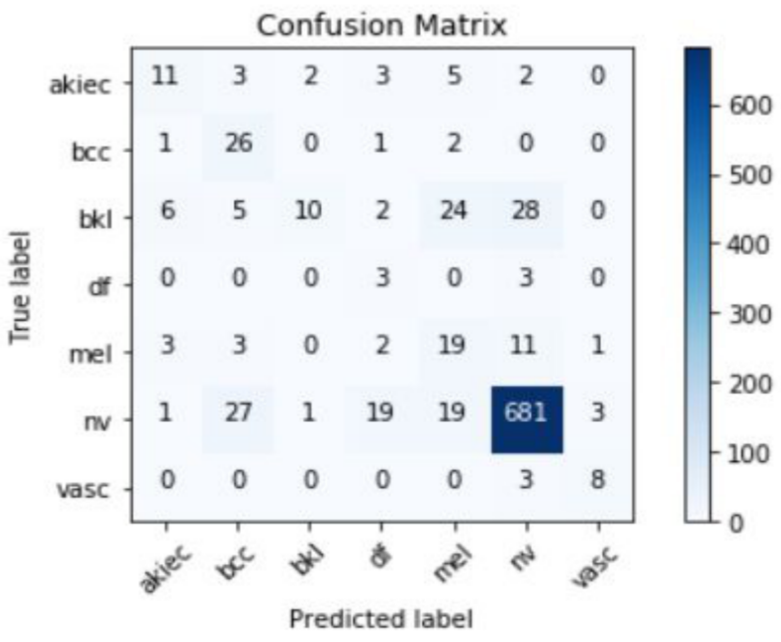


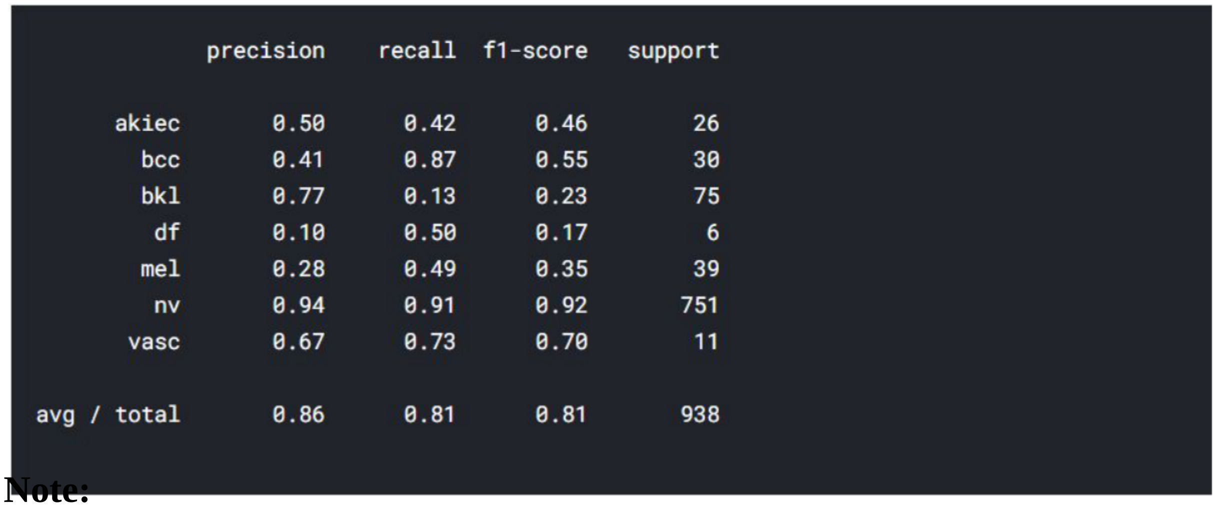






Confusion Matrix Generation:





```
             precision    recall  f1-score   support

      akiec       0.50      0.42      0.46        26
        bcc       0.41      0.87      0.55        30
        bkl       0.77      0.13      0.23        75
         df       0.10      0.50      0.17         6
        mel       0.28      0.49      0.35        39
         nv       0.94      0.91      0.92       751
       vasc       0.67      0.73      0.70        11

avg / total       0.86      0.81      0.81       938
```

Note:

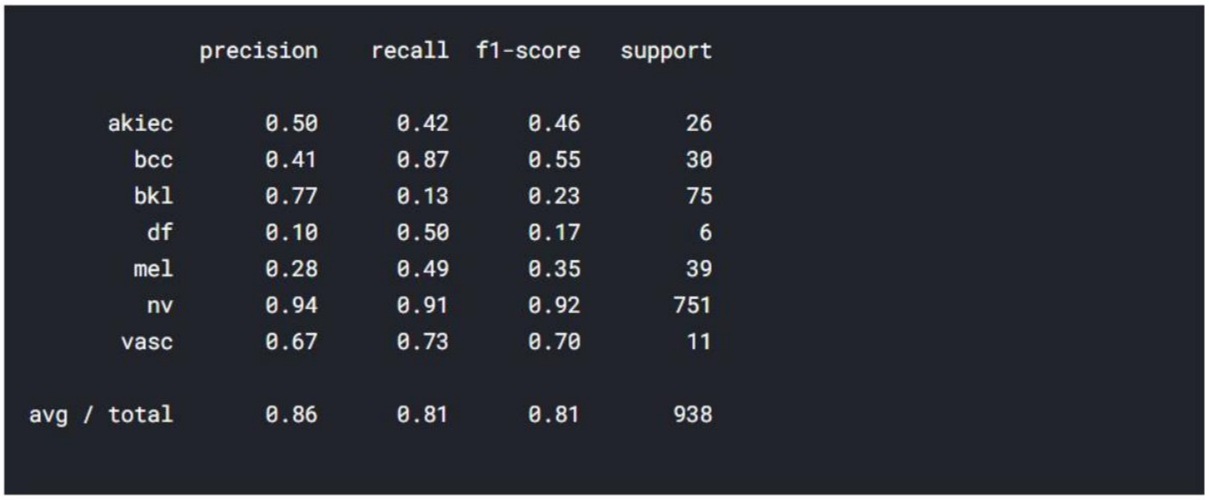
Complete code can be found in the following notebook:
https://colab.research.google.com/drive/1B0utyR8BdwM_Qin_sWrRxzCMssVytH1m?authuser=1
(open in Google colab and use sastra mail id).

Conclusion

Finally by using the DCNN model we successfully build a model and trained it such that we can classify the melanoma images from non cancerous images.
Our model was run on HAM ISIC 2017 Dataset which contained 10,000 images 9,077training, 938 validation and effectively produced significant results which outperformed naïve skin lesion segmentation methods and produced a very good accuracy of 95.7%

- We got an accuracy of 95.74 after classification of images.
- And a top 2 predictions of 96.3 accuracy and 98.6 for top 3 predictions.

Therefore it can be seen that our final results indicate that the accuracy is greater than 91 % using Deep CNN mutual bootstrapping model. The accuracies seen are greater than the diagnosis accuracies that the trained physician obtains as a final result, which range between 30% to 85 % according to literature survey.



```
             precision    recall  f1-score   support

      akiec       0.50      0.42      0.46        26
        bcc       0.41      0.87      0.55        30
        bkl       0.77      0.13      0.23        75
         df       0.10      0.50      0.17         6
        mel       0.28      0.49      0.35        39
         nv       0.94      0.91      0.92       751
       vasc       0.67      0.73      0.70        11

avg / total       0.86      0.81      0.81       938
```

Future Plans

So our plan is to outstretch the proposed Deep CNN model for structure of end to end learning. We enhance both the segmentation and classification networks together, which is to not only to just facilitate the way toward training to incredible degree but also to fundamentally improve the intensity of the highlights that are found out by the model. The model we constructed was explicit to skin lesion diagnosis and analysis however further can be stretched out to different conventional other profound learning-based clinical picture tasks in the near future.

References

- Yutong Xie†, Jianpeng Zhang†, Yong Xia, and Chunhua Shen (2020) A Mutual Bootstrapping Model for Automated Skin Lesion Segmentation and Classification.

- J. Zhang, Y. Xie, Q. Wu, and Y. Xia, “Skin lesion classification in dermoscopy images using synergic deep learning,” in MICCAI, 2018, pp. 12–20.

- https://www.tensorflow.org/

- https://www.analyticsvidhya.com/blog/2018/12/guide-convolutional-neural-network-cnn/

- https://deeplizard.com/

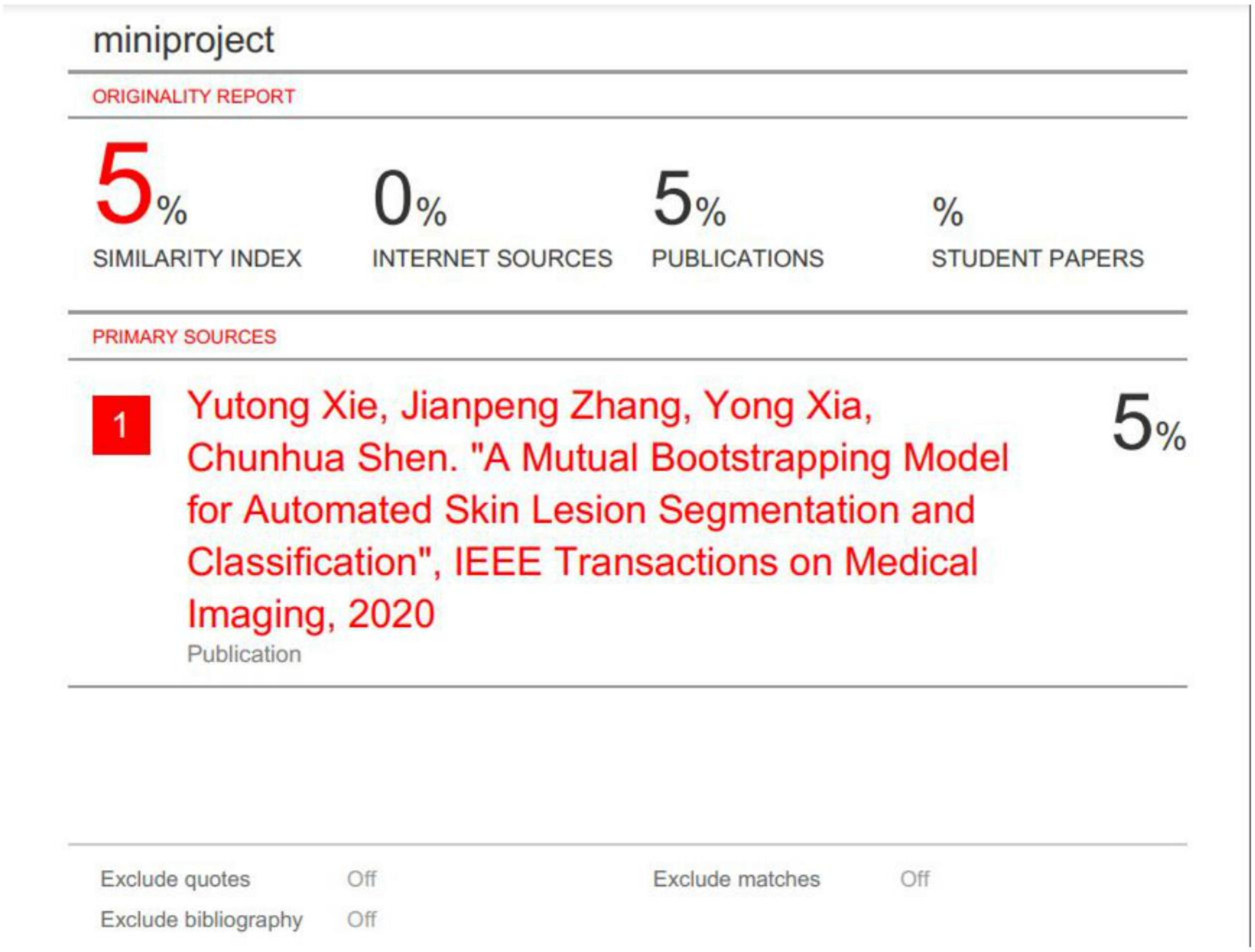
- https://keras.io/

- R. Amelard, J. Glaister, A. Wong, and D. A. Clausi, “High-level intuitive features (hlifs) for intuitive skin lesion description,” IEEE Transactions on Biomedical Engineering, vol. 62, no. 3, March 2015.

- Machine Learning in Medical Imaging" , Springer Science and Business Media LLC, 2019

- Pablo G. Cavalcanti, Jacob Scharcanski. "Chapter 2 Macroscopic Pigmented Skin Lesion Segmentation and Its Influence on Lesion Classification and Diagnosis" , Springer Science and Business Media LLC, 2013

Turnitin Report



miniproject

ORIGINALITY REPORT

5%	0%	5%	%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	Yutong Xie, Jianpeng Zhang, Yong Xia, Chunhua Shen. "A Mutual Bootstrapping Model for Automated Skin Lesion Segmentation and Classification", IEEE Transactions on Medical Imaging, 2020 Publication	5%

Exclude quotes	Off	Exclude matches	Off
Exclude bibliography	Off		