Project Report (September 30, 2021)

Ankita Ghosh[1] and Sahil Khose[2]

[1]Research Assistant, ghoshankita0907@gmail.com , CSE, MIT Manipal
[2]Research Assistant, sahilkhose18@gmail.com, ICT, MIT Manipal

ABSTRACT

In this report we discuss the various research papers that address the problem of measuring severity of age related macular degeneration. We also report the method that we have designed and the results for the same as compared to the other methods. We also thoroughly discuss the future plans that we have for the project.

Literature Review

A systematic review and meta-analysis: Dong and Yang et al

1. Bhuiyan et al, 2020 and Govindaiah et al, 2018
   - AREDS dataset 116,875 images + demographic data
   - Deep Learning classification model of 12 classes + prediction model for risk of progression to late AMD.
   - Preprocess: RBG to L*a*b* $\rightarrow$ DL: ensemble of 6 networks- **Xception, Inception-ResNet-V2, Inception-V3, NasNet** (developed in Govindaiah et al, 2018 and inspired from Grassmann et al, 2018).
   - DL probability values, drusen area quantification, reticular pseudo-drusen (RPD) categorization, AMD category and demographic data passed to prediction model: Logistic Model Tree (LMT)
2. Grassmann et al, 2018
   - AREDS and KORA dataset, 120000 images, 13 classes
   - Processing: fundus images are normalized to have equal illumination and color balance.
   - 6 different convolution neural net models are trained: **AlexNet, GoogLeNet, VGG with 11 convolution layers, Inception-V3,ResNet with 101 layers and Inception-ResNet-V2 (I-ResNet-v2)**.
   - class predictions used to train a random forest classifier to improve classification accuracy.
3. Peng et al, 2019
   - ARDES dataset, 59302 images, 6 classes
   - DeepSeeNet consists of 3 constituent parts, all designed with an **Inception-v3** architecture:
     (a) a sub-network, **Drusen-Net (D-Net)**, which detects drusen in 3 size categories (small/none, medium, and large)
     (b) a sub-network, **Pigment-Net (P-Net)**, which detects the presence or absence of pigmentary abnormalities (hypopigmentation or hyperpigmentation)
     (c) a sub-network, **Late AMD-Net (LA-Net)**, which detects the presence or absence of late AMD (neovascular AMD or central GA)
     Combination of the outputs from these networks used to calculate final classification
   - Preprocessing as stated in Burlina et al, 2017
4. Burlina et al, 2017
   - AREDS dataset, total 5664 images, 4 classes. Previous paper: (a) 2017 Nov uses AlexNet for binary classification. (b) 2011 detects anomalies and exudates in retinal image using one-class classifer based on **SVM**.

- thresholding to detect and crop the outer boundaries of the retina → resizing to pass to OverFeat DCNN → Features of the OverFeat DCNN architecture used and feature vector normalized → Linear SVM
- Multigrid processing: Several crops from an image concatenated and passed DCNN

5. Kankanahalli et al, 2013

    - AREDS dataset, total 2772 images, 4 categories and 3 class classification
    - 'bag of words' concept used: text = fundus images and visual words = visual features computed in these fundus images
    - ROI obtained using channel extraction, median filters, morphological transforms etc
    - keypoint and feature extraction: SURF, colorspace manipulation (L*a*b*)
    - vocabulary creation: centroids found using **K-Means clustering**

6. Mookiah et al, 2015

    - Private, ARIA and STARE dataset, 784 images, 4 classes. Previous papers: 2014 Sept, 2014 Oct perform similar techniques
    - Feature extraction: LBP (Local binary pattern) and LCP (Local configuration pattern) algorithms used
    - Models: **Decision Tree, Naive Bayes K-NNs, SVMs**

```
---------------Dataset---------------
IDRiD Testing Set  : 103
IDRiD Training Set  : 413
drive  : 31
messidor2  : 80
Total datapoints:  627

----------------Labels----------------
Label  0 :       295
Label  1 :       58
Label  2 :       274
Total datapoints:  627
```

Figure 1. Our Data Statistics

Our Methodology

- **ResNet-18** (11M parameters) is implemented to be trained for this 3-class classification problem.
- We use pre-trained **ImageNet** weights and train the entire network over the dataset.
- The optimizer that we have used is **Adam optimizer** with a learning rate of **5e-4**.
- The data was **over/under sampled** during batch generation to balance the **class imbalance** mentioned in previous report.
- The batch size found optimal is **64**.
- The suitable architecture, optimizer, batch size and learning rate was determined after performing logarithmic hyperparameter search over **3 dozen** different combinations.

Method	Accuracy	No. of datapoints	Classes
Ours	**88.10%**	627	3
Burlina et al	79.4%	5664	4
Govindaiah et al	86.1%	116,875	4

Table 1. Classification Results - 1

Method	Accuracy	No. of datapoints	Classes
Ours	88.10%	627	3
Kankanahalli et al	81.8%	2772	3
Mookiah et al	90.2%	784	4
Bhuiyan et al	**96.1%**	116,875	4

Table 2. Classification Results - 2

Results

- Table 1 demonstrates our results (accuracy) being **superior** to other similar methods with respect to number of classes.
- Table 2 demonstrates our results compared to **other superior methods**. The main reason for these methods being superior seems to be the **amount of labelled data available**.
- We have the **lowest number of datapoints** amongst the 14 papers that are mentioned in: **A systematic review and meta-analysis: Dong and Yang et al**
- Table 3 shows the **highest number of datapoints** in descending order in the 19 paper survery as compared to us.
- This demonstrates us the urgency of availability of high number of datapoints

Discussion

- We have our pipeline and benchmark ready for classification.
- Further we will be working to get more data which clearly will help our results as in 3.
- We are also looking to modify our architecture to reduce overfitting and improve our results.
- 3 future aspects: **1. Semi-supervised Learning, 2. Self-supervised Learning and 3. Multi-task Learning**.

Method	No. of datapoints	Classes	Accuracy
Ours	627	3	88.1%
Zapata et al	306,302	2	86.3%
Gonzalez-Gonzalo et al	134,421	2	85.9%
Burlina et al	133,821	2	91.6%
Grassmann et al	120,656	13	63.6%
Bhuiyan et al	116,875	4	96.1%
Govindaiah et al	116,875	4	86.1%
Ting et al	108,558	2	88.8%
Keenan et al	59,812	2	96.5%
Peng et al	59,302	6	67.1%
Keel et al	56,113	2	96.5%

Table 3. High number of datapoints

Semi-supervised Learning & Distillation

- **Semi-supervised learning:** Collecting and using vast amount of unlabelled data by generating pseudo-labels
- We have the **pipeline ready** for 2 algorithm from Khose et al paper and we will be using the same for experimentation
- **Knowledge Distillation** has also been significantly helpful for semi-supervised methods Khose et al
- We have the **pipeline ready** for the same, and will be used for experimentation for our use case.

Algorithm 1: Semi-supervised classification train loop

Input: Sample image
Output: Class of the given image
for $epoch \leftarrow 0$ **to** E **do**
    if $epoch < E_i^\alpha$ **then**
        $\alpha \leftarrow \alpha_i$
    else if $epoch < E_f^\alpha$ **then**
        $\alpha \leftarrow \frac{\alpha_f - \alpha_i}{E_f^\alpha - E_i^\alpha} * (epoch - E_i^\alpha) + \alpha_i$
    else
        $\alpha \leftarrow \alpha_f$
    end if
    Run the model on train set
    $loss \leftarrow BCE(l, \hat{l}) + \alpha * BCE(u_{epoch}, u_{epoch-1})$
    Generate the pseudo labels for unlabeled data
    Evaluate the model on validation set
end for

Figure 2. Semi-supervised algorithm

Self-supervised Learning

- This refers to pre-training of network without using any labels, generally by using **contrastive loss methods**
- Many papers like SimCLR-V2, MOCO, CPC-V2 have showed that this pretraining method **outperforms the previous fully supervised** benchmarks.
- These SSL methods have also seemed to be extremely effective in **histology**
- Dehaene et al using the architecture in 3 proves this by implementing this technique and massively improving the results by **6% on the previous state of the art benchmark**.

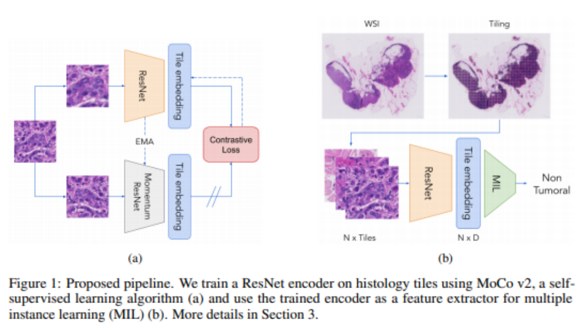

Figure 3. Self-supervised architecture

Multi-task Learning

- Training **multiple tasks** using a **singular backbone** by sharing weights for simultaneous training has been very helpful in low datapoint cases.
- This has already been used in many medical DL papers like:
  - 1. Le et al having the 4 architecture for **cancer diagnosis**.
  - 2. Amyar et al having the 5 architecture for **COVID diagnosis**.
  - 3. He et al having the 5 architecture for **Thoratic Organs Risk Analysis.**

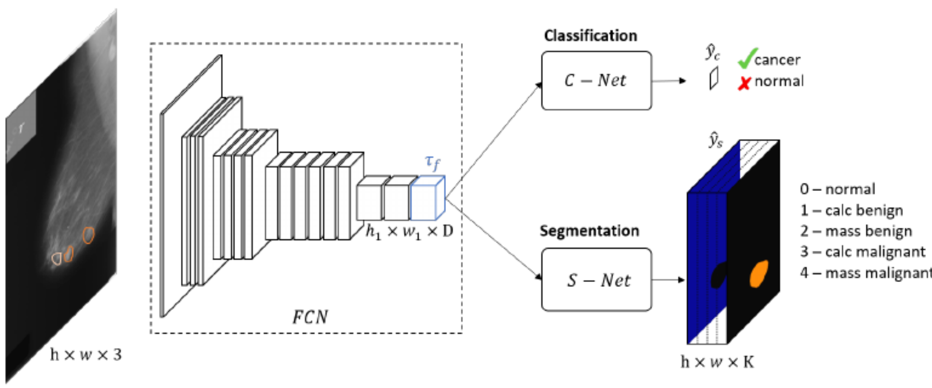

Figure 4. Cancer Diagnosis

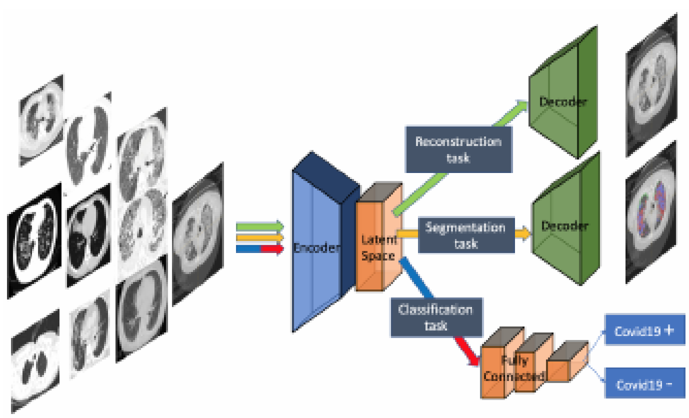

Fig. 4. Our proposed architecture, composed of an encoder and two decoders for image reconstruction and infection segmentation. A fully connected layers are added for classification (COVID vs non-COVID)

Figure 5. COVID Diagnosis

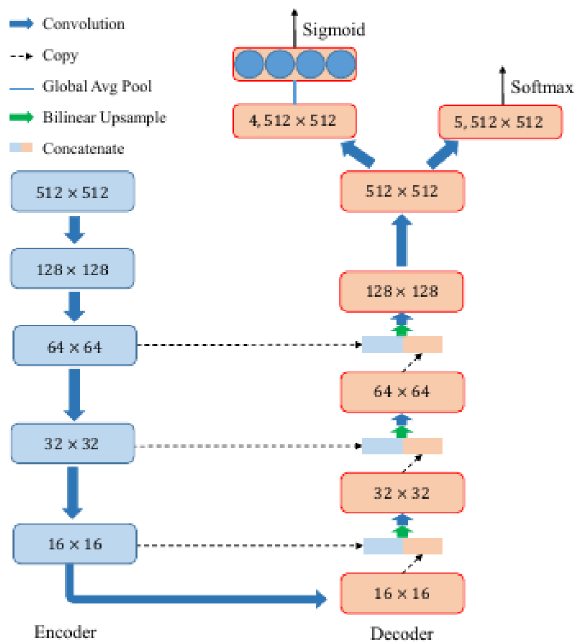

Figure 6. Thoracic Organs