Inappropriate Image Detection with Convolutional Neural Networks and Transfer Learning

Szapula László, Ulicska Gergely Ádám
Budapest University of Technology and Economics
Budapest, Hungary
{laszlo.szapula, ulicskagergo}@edu.bme.hu

***Abstract* - Nowadays so many have easy access to the Internet - that is a good opportunity and many times dangerous as well. Underaged can find sensitive or even insensible pornographic content in a minute because of the almost lack of censorship on most sites. Lot of these do not even have any checking algorithms. In this paper we have an objective to find out from a picture if it contains the so-called Region of Interests (ROI). We built a deep neural network on this purpose with transfer learning and fine tuned the whole. The output of our algorithm is a binary classification: appropriate or inappropriate. We managed to demonstrate the network with an application and a server as well.**

Keywords - pornography, nudity, appropriate, inappropriate, region of interest, image processing, deep learning, convolutional neural networks, adault image classification, fine tuning

I Introduction

Motivation. After hours of looking for papers in this subject we have not really found any papers that use deep convolutional networks (CNNs) for detecting underclothing. Of course, there are many researches in nudity detection, but underclothing is a subject that is not really researched. Because of that we chose nudity detection papers as our motivation. One such paper is [1], which detects pornography both in images and videos. [2] does not use deep learning, but uses a mathematical approach to detect skin areas on an image. We wanted to use this as a preprocessing method before we feed the pictures in our convolutional network, but because of the shortness of time we have not used it.

Objective. Our goal is to build a kind of network that can consistently tell if a person in the picture wears too few clothes or if these clothes are too inappropriate. The objective is a suitable convolutional network with the proper classifier that can do this work. We wanted to get the best accuracy, so we tested many existing pre-trained networks (described in section II), tried different architectures for the classifier and fine-tuned the finite network with hyperparameter optimization.

Results. We basically created a deep convolutional neural network that can predict very effectively, if the person in the picture wears enough clothes. The network can even detect such scenarios, that the person wears partially transparent colored clothes. We even made an android application that sends the picture to our server, and tells, if the picture is appropriate.

II Data collection

We decided not to use given data collections. We wanted to train our network on brand new data. Thus we downloaded our database from the internet[1]. As it is described in the results section we trained our network only with pictures containing females. We searched on the sites Google and Pinterest for two categories of females: appropriate and inappropriate and downloaded around 3000 pictures to build a significant database but we had to sort these because not all were proper. We performed this sorting with a python script and then manually as well. At the end of the sorting we had 2365 images, which shouldst be divided into three groups (folders) for the network: train, validation and test database must be built. We worked with 70-20-10 percentages. All the folders contain two categories of images: appropriate and inappropriate 50 percent of each.

[1] With the extension on Google Chrome named Imageye (can be found here)

III Model choice

First we had to find out which pre-trained model we will use in the transfer learning as a base model. We tested four very different alternatives: InceptionV3[2], ResNet101V2[3], NASNetLarge[4] and VGG19[5]. To accomplish this step we had to make some preconditions:

- the dataset contained 1572 train, 527 validation, 266 test pictures;
- layers on the base model were a maxpooling2d to reduce the size of the pictures, a dropout with 0.25 percent chance to skip a neuron, a fully-connected dense layer with 1024 neurons and relu activation, another dropout with 0.25 percent and one more fully-connected layer as output;
- top layers of the base models were not included as we accomplished transfer learning;
- as optimizer we used adagrad, because the pre-measurements showed this better than adam, but in the hyperparameter optimization section adam turned out the best;
- callbacks: early stopping to prevent overfitting (with patience of 10 and 0.001 minimum delta time) and checkpointer to save the best model;
- in the fitting step we worked with 29 steps per epoch, 10 validation steps and 5 epochs.

Model name	Before freezing				
	Time	loss	acc	val_loss	val_acc
Inception V3	243	0.0669	0.9752	0.1167	0.9692
ResNet101V2	194	0.3243	0.9669	0.4613	0.9327
NASNetLarge	257	0.1250	0.9739	0.2312	0.9577
VGG19	179	0.4837	0.7684	0.4138	0.8000

After freezing					test_err	test_acc
Time	loss	acc	val_loss	val_acc		
249	0.0537	0.9771	0.1496	0.9500	0.1454	0.9511
214	0.1120	0.9720	0.2541	0.9385	0.2398	0.9436
408	0.1107	0.9726	0.5400	0.9173	0.2049	0.9624
180	0.4741	0.7729	0.4112	0.8038	0.4139	0.8120

TABLE I. Comparison of base models (time measured in sec)

[2] https://keras.io/api/applications/inceptionv3/
[3] https://keras.io/api/applications/resnet/
[4] https://keras.io/api/applications/nasnet/
[5] https://keras.io/api/applications/vgg/

As we can see in the final results, the InceptionV3 turned out to be the best base model for our dataset, since it showed the best error and accuracy on the test dataset. Among the other models, it is apparent that InceptionV3 had the best results before and after freezing too. We considered the use of the NASNetLarge as a base model as it had a better test accuracy but as the time consumption of it was very high[6] and beneath it had a significant bigger test error as well, we did not choose it after all.

Based on these measurements henceforward we worked with the InceptionV3 model and fine tuned the whole model's hyperparameters as follows.

IV Network architecture

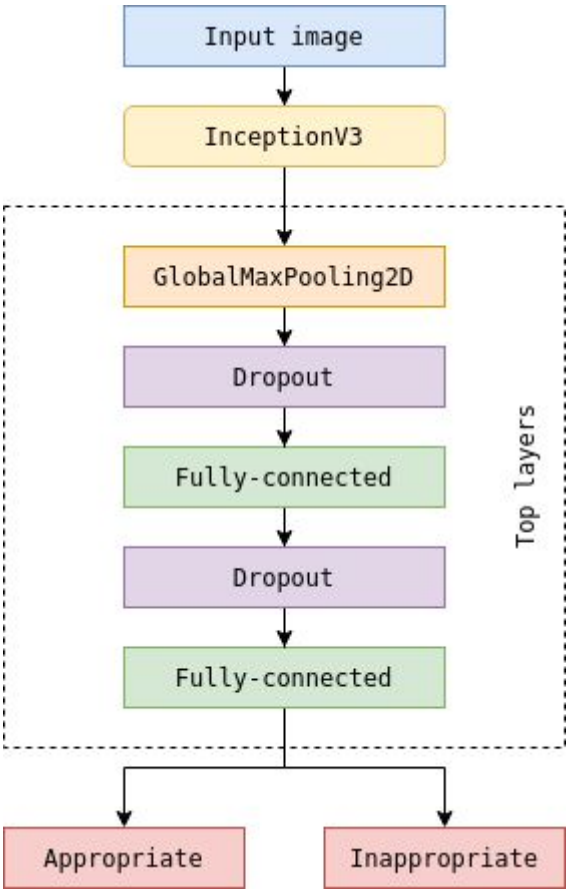


FIGURE I. Network architecture

As described in the previous section, we used the InceptionV3 pre-trained model as a base model. It is important to pay attention to the size of the input images: they must be 299x299 pixels[7]. After InceptionV3 which top layers were not included we built a top architecture. First we reduced the size of the pictures with the GlobalMaxPooling2D layer, then we used two dropout and two fully-connected (dense) layers. These layers can be optimized and fine-tuned in the next section with the hyperparameter optimization. Only the units of the second fully-connected layer is bounded, it must be one as our objective is a classification problem with two classes. This last dense layer's output is either appropriate (there is not any ROI on the input picture) or inappropriate (the network has found that the picture contains seamless regions).

V Hyperparameter optimization

Hyperparameter optimization is the most important step after building our network. With this optimizer we can earn the best possible performance of our CNN. There are many frameworks for automatic optimization, but we chose Tensorflow's Keras Tuner[8] module. For the shortness of time, we picked 4 hyperparameters to fine tune our network: the two rates of the two Dropout layers, the activation of the hidden Fully-connected layer and the learning rate.

For the values of the dropouts we tested [0.0, 0.1, 0.15, 0.25, 0.3, 0.45, 0.5], from which 0.25 was the best for the first Dropout layer, and 0.1 for the second. As for the activation of the hidden Fully-connected layer, we tried ['swish', 'relu', 'sigmoid'] functions. Of course, ReLU won the contest here. For the learning rate we only tried two values: 0.01 and 0.001. In an interesting manner, 0.01 was the optimal choice.

With these settings Keras Tuner could achieve a validation accuracy of 0.9712 after 5 epochs of training on our CNN. Therefore we used these settings for the further training of our model:

1. First dropout rate of 0.25;
2. Second dropout rate of 0.1;
3. Fully-connected layer's activation: ReLU;
4. Fully-connected layer's learning rate: 0.001.

VI Mobile application and server

To demonstrate the usage of our CNN, we made a simple application for android mobiles. With this application we can take a picture with the camera of the mobile or select an image from the gallery and send it to our cloud based server for analysis. After the evaluation of the image the server sends back a response and based on that the application tells us if the image is appropriate or not.

The server is a python flask application, which uses Keras to load our previous described model from an input file and propagate the received image through the network. The image can be sent by a simple POST request in the body with the 'image' key. Then based on the output of the network sends back the server the decision in JSON format, where the 'answer' attribute contains the prediction. The server is running on heroku[9]. Unfortunately, the response time is quite slow yet, it takes around 3-4 seconds. This comes with the limited resources of the free plan in heroku, but in the future we would like to enhance the performance.

VII Results

As we setted up the final parameters received from the hyperparameter optimization, we had got the final results as follows:

The loss after the unfreezing was 0.0562, the accuracy stopped at 0.9796. As we measured the accuracy metrics we received a 0.2750 validation loss and 0.9596 validation

[6] As it has 1244 layers, it is comprehensible.

[7] For ResNet101V2 it is 224x224, NASNetLarge must have 331x331 or 224x224, for VGG19 the size is also 224x224.

[8] https://keras-team.github.io/keras-tuner/

[9] https://www.heroku.com

accuracy at the end of the training. We calculated the test error and accuracy of our model as well and these were 0.1116 and 0.9549. It is apparent that these are not the best values but we think we could achieve better results with the methods described in the following section in the future.

As it is known the data measurements are worthless without a proper analysis. We accomplished this with a confusion matrix beside the test database; it is shown below. As it is apparent we achieved good results, the true values are significantly higher compared to the false values.

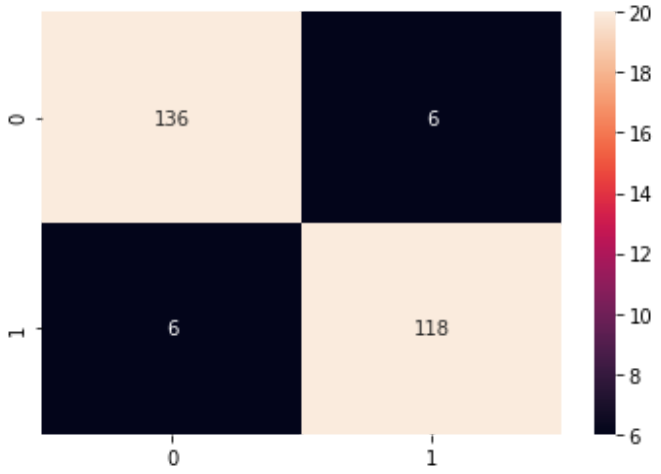


FIGURE II. The confusion matrix[10]

From the confusion matrix we calculated the following ratios to evaluate our network:

The sensitivity (means how many relevant items are selected) is the proportion of true positives and all the positives so it is 118/(118+6) = 95.1613%.

The specificity (means how many negative elements are truly negative) is the true negatives divided with all the negatives so 136/(136+6) = 95.7746% which is a little better than the sensitivity.

We claim that in this field the specificity is more relevant because we do not want the so-called “false alarm” problem that is why we think these results are good.

With this network we made an interesting experiment as well, as we fitted the whole model only with pictures containing females. We wondered if it could classify pictures of males. The answer is yes, it is capable of classifying any gender.

Here follows four test data of the classes of the tests: we tested females and males with appropriate and inappropriate clothing as well. In the table we can see that the pictures have got different sizes, our network can deal with all of them as apparent.

[10] The confusion matrix was created with the math library of tensorflow and plotted with seaborn.

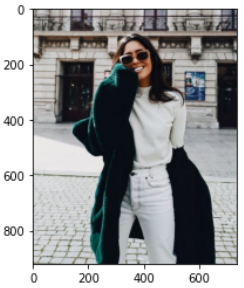
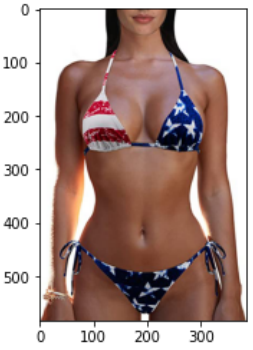
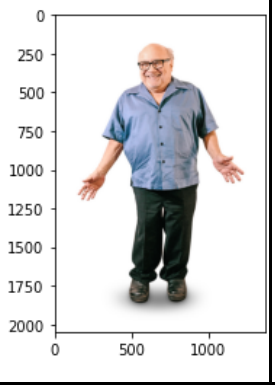
	Appropriate	Inappropriate
Female		
Male		

TABLE II. Test results for the four test cases

We tested our network on critical images as well when it is not apparent first that the picture is inappropriate. As for example the following picture:

FIGURE III. A critical inappropriate picture

In this image there are ROIs and the dress is also very smooth and transparent but for the first glance it is appropriate. That is why we were curious if the model can classify this as well. Fortunately it could achieve this and other indifferent pictures and classify them as inappropriate.

VIII Future work

Because of the shortness of time we could not add more enhancements to our model but this is not a big problem, we

would like to enhance our network in the future as well. We have got some plans for this work:

First of all we would like to add some preprocessing features to our network like described in article [2]. This may conduct a better result and will be a way faster.

Another top layer also could be better in our transfer learning, we could achieve better loss value. This could be fine tuned with the hyperparameter optimization as well.

At the time we have got an appropriate network we would like to decrease the false positive ratio (proportion of the false positive examples and all the negatives: false positives and true negatives), because we suppose that in this field the “false alarms” - that the application claims that the picture is not appropriate - is a bigger problem than it states that the image is appropriate when not.

IX Related work

As we began to construct our image classifier network we searched for other implementations as well. Among these we would like to highlight paper [2] as it does not contain deep learning methods but with the skin detection we could achieve better results and this was a good starting point for us.

The [1] addresses the issue with images and videos as well. The method is not so complicated, only the videos are cut into separate images and the network deals with separated pictures. In the future we could manage something like this. They work with transfer learning as well with two large base models: AlexNet and GoogLeNet and combine these into a so-called AGNet.

In the paper [4] the authors concentrated on videos but created a convolutional neural network to work with as we have done. They had very good results, around 98-99 percent. In [5] the approach was based on an adversarial network, we could try one in the future as well.

In article [3] the authors worked with colored-skin detection but the results were not the best, not even accuracy (around 80%). In [11] there is a good introduction to morphological operators for skin detection so maybe we could use them.

[10] describes an algorithm for this problem without deep learning but these methods are a good comparison when we have the final results. [8] is also a non-deep learning approach but describes the color spaces for skin detection very well.

X Conclusion

We had the object to compose a convolutional neural network to classify appropriate and inappropriate images. As the results show we achieved this object well but there is still room for improvement. As described we plan to work with our network in the future, we would like to add a preprocessing skin detection to it, but as apparent we built a very simple model and this works also well on these test data.

We are continuing the research in the field of skin detection with deep learning as well. Beside we would like to create a method for video classifying.

The results show also that our experiment to teach the network only with data containing females and testing on males as well was also successful.

We are proud that we could achieve an application with a server as well to demonstrate our network in progress. We could upgrade this system because as it is described in the sixth section it is not the fastest yet.

We expect that all the social media platforms will apply some methods like these on their sites to filter inappropriate pictures and protect their underaged users.

References

[1] Mohamed Moustafa. 2015. Applying deep learning to classify pornographic images and videos.

[2] Mirza Rehenuma Tabassum, Alim Ul Gias, Md. Mostafa Kamal, Hossain Muhammad Muctadir, Muhammad Ibrahim, Asif Khan Shakir, Asif Imran, Saiful Islamm, Md. Golam Rabbani, Shah Mostafa Khaled, Md. Saiful Islam, Zerina Begum. 2010. Comparative Study of Statistical Skin Detection Algorithms for Sub-Continental Human Images

[3] Manuel B. Garcia, Teodoro F. Revano, Jr., Beau Gray M. Habal, Jennifer O. Contreras, John Benedict R. Enriquez. 2018. A Pornographic Image and Video Filtering Application Using Optimized Nudity Recognition and Detection Algorithm

[4] Pedro V. A. de Freitas, Paulo R. C. Mendes, Gabriel N. P. dos Santos, Antonio José G. Busson, Álan Livio Guedes, Sérgio Colcher, Ruy Luiz Milidiú. 2019. A multimodal cnn-based tool to censure inappropriate video scenes

[5] Martin D. More, Douglas M. Souza, Jonatas Wehrmann, Rodrigo C. Barros. 2018. Seamless Nudity Censorship: an Image-to-Image Translation Approach based on Adversarial Training

[6] Paulo Vitorinoa, Sandra Avilab, Mauricio Perezc, Anderson Rocha. 2017. Leveraging Deep Neural Networks to Fight Child Pornography in the Age of Social Media to Fight Child Pornography in the Age of Social Media.

[7] Gabriel S. Simoes, Jonatas Wehrmann, Rodrigo C. Barros. 2019. Attention-based Adversarial Training for Seamless Nudity Censorship.

[8] Rahat Yeasin Emon. 2020. A Novel Nudity Detection Algorithm for Web and Mobile Application Development.

[9] Aloisio Dourado, Frederico Guth, Teofilo de Campos, Li Weigang. 2020. Domain Adaptation for Holistic Skin Detection.

[10] Rigan Ap-apid. 2005. An Algorithm for Nudity Detection.

[11] Alessandra Lumini, Loris Nanni, Alice Codogno, Filippo Berno. 2018. Learning morphological operators for skin detection.

[12] M. R. Tabassum, A. U. Gias, M. M. Kamal, H. M. Muctadir, M. Ibrahim, A. K. Shakir, A. Imran, S. Islam, M. G. Rabbani1 , S. M. Khaled, M. S. Islam, Z. Begum. 2010. Comparative Study of Statistical Skin Detection Algorithms for Sub-Continental Human Images.

[13] Danilo Coura Moreira, Joseana Macêdo Fechine. 2018. A Machine Learning-based Forensic Discriminator of Pornographic and Bikini Images.

[14] Idoko John Bush, Rahib Abiyev, Mohammad Khaleel Sallam Ma’aitah, Hamit Altıparmak. 2018. Integrated artificial intelligence algorithm for skin detection.

[15] A. A. Zaidan, H. Abdul Karim, N. N. Ahmad, B. B. Zaidan And A. Sali. 2013. An automated anti-pornography system using a skin detector based on artificial intelligence: a review.