CS 231N Final Project: Deep Visual Learning of Reddit Images

Tyler Chase
Stanford University
tchase56@stanford.edu

Rolland He
Stanford University
rhe@stanford.edu

Kareem Hegazy
Stanford University
khegazy@stanford.edu

Abstract

Recent advancements in CNN architectures have seen marked improvements for image classification, such as in the ImageNet Large Scale Visual Recognition Challenge (ILSVRC). In this paper, we investigate the performance of 3 architectures – AlexNet, GoogLeNet, and ResNet – on Reddit data. Two classification tasks are investigated: predicting the subreddit community an image is from, and predicting if an image is safe-for-work (SFW) or not-safe-for-work (NSFW). We find that the highest performing models from best to worse in almost every case is as follows: GoogLeNet, ResNet, and AlexNet. Since there is significant correlation between subreddits and NSFW image tags, we employed multitask learning and found almost across the board improvements for both tasks. Lastly, we have generated saliency maps to qualitatively assess the performance of our best-performing model; we find that the key features are able to be identified in images from many different subreddits.

1. Introduction

1.1. Background

Reddit is an online social news aggregation and internet forum. With over 540 million monthly visitors, 70 million submissions, and 700 million comments [1], Reddit offers a rich dataset for various analyses. The site rewards interesting posts and users who submit them in the form of ”karma”, given by others in the form of upvotes. The site is also sectioned into various subcommunities, called ”subreddits”, each of which focuses on different topics, in which users post relevant content.

1.2. Problem

Individual subreddits are often devoted to a particular topic or theme (like food or space, for example). Therefore, images from any given subreddit tend to be quite related in their content, making it possible to differentiate between content from different subreddits. At the same time, there are also many subreddits that encompass broader themes such as nature or animals, which can often be confounded with each other. An interesting problem, therefore, is to predict if an image would fit well in a subreddit community. Due to the possibility of similar features between subreddits (i.e. animals and nature), such a classification algorithm will have to learn more then simply object recognition, but what specific features or image styles separate such similar categories. This sort of algorithm could have application in connecting content creators to subreddit communities.

In addition, some submitted images on Reddit contain adult content (i.e. pornography). These images are required to have an NSFW (not safe for work) tag, so that they are hidden from any user who has not explicitly enabled NSFW images. However, users can sometimes forget to tag their own posts as NSFW, in which case moderators need to tag the post themselves. Due to the delay of human moderation, NSFW images can occasionally go untagged for some period of time, showing up in the feeds of those who may not want to view such images. Therefore, having a real time algorithm that can identify posts as NSFW can be quite useful for either notifying a moderator, or tagging the image itself.

Lastly, multitask classification [2] is a method that allows a model to simultaneously learn multiple tasks. This has been shown to improve performance on each individual task when there is a correlation between the tasks. Since many subreddits do not allow adult content, while others exclusively host adult content, we seek to determine if multitask learning leads to improvements in both subreddit and nsfw classification.

1.3. Related Work

While our subreddit classification task shares characteristics with the ImageNet challenge in that we attempt to categorize images, we focus on relevance to a subreddit community, which often encompasses broader themes than single objects. For example, the subreddit r/Art or r/Creepy

[1] http://www.redditblog.com/2015/12/reddit-in-2015.html

encompass abstract concepts that have drastic differences in their manifestations when applied to different objects. Moreover, multiple subreddits can share many primary features, as mentioned above, unlike ImageNet. Such a sharing of features requires the algorithm to learn more abstract concepts about the relationship between such features. To the best of our knowledge, there has not been any similar work done on Reddit data before, so our work presents a novel application of deep visual learning.

2. Approach

2.1. Objective

We use three CNN architectures to classify images into the their correct subreddit and tag them NSFW. We chose AlexNet as a baseline model, as well as GoogLeNet and ResNet as our more advanced models. For each architecture, we train three separate models: one for standalone subreddit prediction, one for standalone NSFW prediction, and lastly one for using multitask learning to classify both the subreddit and NSFW labels. The results for all these trainings were compared across the three architectures.

After training and tuning our models, we take our model with the highest validation accuracy and create saliency maps for image visualization. The goal of this is to qualitatively asses if the algorithm identifies characteristics we consider important for a given subreddit or NSFW tag.

2.2. Dataset

2.2.1 Reddit Data

This investigation uses the full Reddit Submission Corpus[2], which contains data from all reddit submissions (both posts and comments) categorized by subreddits since 2008. Because most subreddits contain either primarily non-image posts or generic images, we only consider 20 hand-selected subreddits with exclusively photo submissions. Moreover, most of the subreddits we have picked were chosen to contain mutually exclusive content; at the same time, we have also included a few very similar subreddits to see if our models can discover more subtle features for distinction. The subreddits investigated are listed in Table 1. The data for each post contains the image URL, the subreddit in which the image was posted, and a boolean representing the existence of an NSFW tag.

Subreddit	Pictures of...	Data Size
r/EarthPorn	nature	1258
r/SkyPorn	the sky	1225
r/spaceporn	space	1119
r/MilitaryPorn	military	1262
r/GunPorn	guns	1171
r/carporn	cars	1258
r/CityPorn	urban areas	1201
r/ruralporn	rural areas	606
r/ArchitecturePorn	architecture	1066
r/FoodPorn	food	1233
r/MoviePosterPorn	movie posters	1309
r/ArtPorn	art	1278
r/RoomPorn	rooms	1274
r/creepy	unsettling images	1215
r/gonewild	undressed women	951
r/PrettyGirls	attractive women	1233
r/ladybonersgw	undressed men	848
r/LadyBoners	attractive men	1143
r/cats	cats	1267
r/dogpictures	dogs	1295

Table 1. List of the 20 subreddits we used, along with short descriptions of their content

2.2.2 Data processing

For data preprocessing, we downloaded and resized the images to 128×128 in RGB format using a Lanczos downsampling filter. In total, we have 31,813 images represented as pixels across three channels. Therefore, our input X is a $31,813 \times 128 \times 128 \times 3$ matrix. For the subreddit classification task, our output y is the vector of subreddit labels. For the NSFW tagging task, y is a binary vector representing whether the image is adult content or not. For training and testing purposes this dataset is split into training, developement, and testing sets using an 80-10-10 split.

2.2.3 Data Statistics

In order to choose posts most relevant to each subreddit, the top 75 posts (by number of upvotes) per subreddit per month from 2015 to 2016 where chosen. Due to image downloading errors or invalid/non-RGB images, the data is not perfectly balanced over each subreddit. The percentage contribution of each subreddit to the entire dataset ranges from 3.6-5.4%. Secondly, due to correlations between subreddit community and NSFW tags, it was impossible to balance both the subreddits and NSFW tags at the same time; consequently, the dataset that we use for multitask learning must be unbalanced in either the subreddits or NSFW tags. Moreover, since we wanted to draw comparisons between the performance of multitask learning, we had to use the same dataset for multitask learning and the individual standalone tasks. Therefore, we settled on using the entire unbalanced dataset, which contained 8.5% NSFW images. Any accuracy results with respect to NSFW classification will therefore naturally be high.

[2]obtained from `http://files.pushshift.io/`

2.3. Architectures

2.3.1 AlexNet

The first architecture that we analyzed for the two classification tasks described above was AlexNet. AlexNet, developed by Krizhevsky et. al., was the first convolutional neural network winner of the ILSVRC and won in 2012 [6]. Convolutional neural networks have much fewer connections and parameters compared to feedforward networks, and are thus easier to train. This architecture contains five convolutional layers and 3 fully connected layers; at the time, the depth was largely limited by the amount of memory available on the GPUs.

2.3.2 GoogLeNet

GoogLeNet was an architecture developed in 2014 by Szegedy et. al. and won the ILSVRC challenge in the same year [9]. The original network was 22-layers long and consisted of stacked inception modules, which themselves are concatenations of several different convolution and pooling layers. GoogLeNet popularized the use of very deep neural networks and experimented with stacking multiple small, but effective modules on top of each other. In 2015, the Szegedy and Ioffe introduced the idea of batch normalization [5], which accelerates the learning of deep neural networks. Moreover, they applied batch normalization to a revised version of their original GoogLeNet model, which led to a noticeable boost in performance. Another notable modification is the removal of the dropout layer at the end of the architecture. The authors state that batch normalization already acts as a regularizer, reducing or even eliminating the need for dropout. More details about their modified architecture can be found in [5].

2.3.3 ResNet

ResNet [4] was developed by Kaiming He et.al. in 2015 and won the ILSVRC challenge that year. ResNet's 152 layers was a drastic change, as no previous network passed 25 layers in depth. ResNet was inspired by the idea that networks should perform better as they grow in depth, as seen by GoogLeNet. Moreover, one would not expect the network to get worse the deeper it became as it could nullify further layers through learning. In order to accentuate this feature, ResNet is made of many stacked residual learning blocks where the input into the block is summed with the output of the block. This method helps the network to nullify unnecessary blocks by learning that the output should be small compared to the input. Each residual block is made of two 3x3 convolution layers with the same number of filters. There are four groups of stacked residual blocks where a convolution with stride 2 is used at the beginning of each group to shrink the image, while the number of filters used for all the layers in the group is increased. ResNet uses batch normalization after each convolution and before all activations. Regularization comes from the network's deep and thin architecture.

2.4. Multitask Learning

Multitask learning has been used with success in machine learning across a broad range of applications [7]. This is commonly done in deep learning by taking the flattened parameters from the final convolutional layer of a neural network and creating two separate paths of fully connected layers resulting in logits for two separate classification tasks. This type of architecture forces the model to choose parameters that can explain more than one task, which often leads to better generalization [7].

2.5. Ensembling

Ensembling is a technique that aims to combine multiple models into a single meta-model. The idea behind doing this is to incorporate the learned features from all of the ensembled submodels and retain the best ones.

Two popular ensembling schemes that were explored in [3] are majority-vote and weighted average. In majority-vote, the ensemble model simply uses the majority class prediction generated by the individual submodels. In weighted average, the normalized class probabilities outputted by each model are averaged with some user-specified weights; the ensemble model then chooses the class with the highest probability as its prediction. These two schemes have both been shown in the paper to improve upon the best performing submodel across a wide range of tasks. Ensembling is particularly helpful when the individual submodels are distinct enough to learn different features of the task.

2.6. Network Visualization

To qualitatively evaluate the performance of our models, we use saliency maps to see if the models are able to identify key visual features characteristic of subreddit classes. A saliency map is a visualization method introduced by Simonyan et. al.[8]. This is done by taking the derivative of the class score corresponding to the predicted output with respect to the input image. For an RGB image, the maximum of the absolute value of this derivative is taken along the color channel dimension. This results in a heat map image where pixels the algorithm thinks are important in determining the output are brighter.

2.7. Evaluation

We use both accuracy and F1 score as quantitative evaluation metrics of our classification models. In order to see if features we expect to be characteristic of subreddits are recognized by our models, we also use saliency maps as a qualitative measure of how well our models perform.

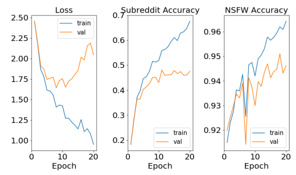

Figure 1. Loss history for AlexNet showing overfitting.

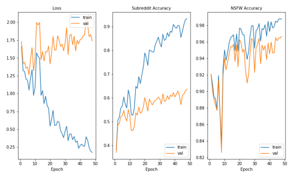

Figure 2. Loss history for GoogLenet showing overfitting.

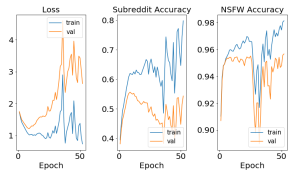

Figure 3. Loss history for ResNet showing overfitting.

3. Experiments

3.1. Model Truncation

For both GoogLeNet and ResNet, we use significantly truncated versions of the full models due to the fact that we only have 20 classes, as opposed to the ImageNet problem (which has 1,000 classes), that these architectures were applied to. Furthermore, our full dataset only contains about 30,000 training examples, far less than that of ImageNet. Thus truncation allows us to reduce the number of parameters in our models since we do not have as much information to train them on, or enough degrees of freedom to make them more unique.

Truncation of both of these models proved quite useful, providing healthy performance boosts of a few percent on both subreddit and NSFW prediction. Most importantly, truncation stemmed large and rapid overfitting by reducing redundant parameters. Moreover, training time was also significantly reduced, providing an additional benefit.

3.2. Architectures

For each of the following 3 architectures, we trained and tuned 3 different models: one for standalone subreddit prediction, one for standalone NSFW prediction, and one for multitask learning combining both tasks.

3.2.1 AlexNet

In this paper we implement a revised version of AlexNet, with a few small modifications:

- Since we run this architecture on only one GPU we simplify the architecture from the two GPU implementation in the original paper [6].
- Due to the fact that we are dealing with smaller images the filters are diminished in size compared to the original paper [6].
- Due to the fact that we are dealing with much fewer categories and a much smaller dataset the fully connected parameters are reduced significantly from 4094 to 2048.

The architecture layout can be seen in Table 2, while the multitask training history in Figure 1.

3.2.2 GoogLeNet

In this paper, we have implemented this revised version of GoogLeNet, with a few small modifications:

- We have removed the auxiliary prediction layers that generate intermediate predictions that are used when backpropogating. This was removed due to the increase in architectural complexity, with the authors noting that the performance gains are quite minimal.
- Due to our smaller input image sizes compared to ImageNet, our final average pooling layer is smaller.
- The initial 1x1 convolutions before the 192 3x3 convolutions were removed, as doing so helped boost performance

See Table 2 for more details on our full architecture.

The model parameters used for training can be found in Table 3. It should be noted that the effect of dropout was minimal, though it did ultimately increase the performance of the model slightly. Experimentation found that batch normalization was best applied right after each convolution layer. ReLU was applied after each batch normalization.

We experimented with truncating various parts of the model, including individual groups of inception modules and convolution/pool layers outside of the modules. In addition, we also tried decreasing the number of filters used in some of the inception layers. In the end, the model that yielded the best performance was one that only retained the first three inception modules and left everything else the same. The loss history during multitask training can be seen in Figure 2.

3.2.3 ResNet

Truncation due to overfitting was the primary reason for changing the architecture. The network was subsequently

Type	filter size/ stride	depth	output size
Input			128 × 128 × 3
Conv 1	11 × 11/4	96	32 × 32 × 96
Max Pool 1	3 × 3/2		16 × 16 × 96
Norm 1			16 × 16 × 96
Conv 2	5 × 5/1	256	16 × 16 × 256
Max Pool 2	3 × 3/2		8 × 8 × 256
Conv 3	3 × 3/1	384	8 × 8 × 384
Conv 4	3 × 3/1	384	8 × 8 × 384
Conv 5	3 × 3/1	256	8 × 8 × 256
Max Pool 3	3 × 3/2		4 × 4 × 256
Flatten			4096
FC 1			2048
FC 2			2048
FC 3			20/2

type	filter size/ stride	depth	output size
input			128 × 128 × 3
convolution	7x7/2	1	64 × 64 × 64
max pool	2x2/2	0	32 × 32 × 64
convolution	3x3/1	2	32 × 32 × 64
sum input/output		0	32 × 32 × 64
convolution	3x3/1	2	32 × 32 × 64
sum input/output		0	32 × 32 × 64
convolution	3x3/2	1	16 × 16 × 128
convolution	3x3/1	2	16 × 16 × 128
sum input/output		0	16 × 16 × 128
average pool	16x16	0	1 × 1 × 128
fully connected		1	1000
logits		1	20

type	filter size/ stride	output size	depth	#1×1	#3×3 reduce	#3×3	double #3×3 reduce	double #3×3	Pool +proj
input		128×128×3							
convolution	7×7/2	64×64×64	1						
max pool	3×3/2	32×32×64	0						
convolution	3×3/1	32×32×192	1			192			
max pool	3×3/2	16×16×192	0						
inception (3a)		16×16×256	3	64	64	64	64	96	avg + 32
inception (3b)		16×16×320	3	64	64	96	64	96	avg + 64
inception (3c)	stride 2	8×8×576	3	0	128	160	64	96	max + pass through
inception (4a)		8×8×576	3	224	64	96	96	128	avg + 128
inception (4b)		8×8×576	3	192	96	128	96	128	avg + 128
inception (4c)		8×8×576	3	160	128	160	128	160	avg + 128
inception (4d)		8×8×576	3	96	128	192	160	192	avg + 128
inception (4e)	stride 2	4×4×1024	3	0	128	192	192	256	max + pass through
inception (5a)		4×4×1024	3	352	192	320	160	224	avg + 128
inception (5b)		4×4×1024	3	352	192	320	192	224	max + 128
avg pool	4×4/1	1×1×1024	0						
dropout		1×1×1024	0						
fully connected		1×1×1000	1						
softmax		1×1×1000	0						

Table 2. The architectures of the modified AlexNet (top left), ResNet (top right), GoogLeNet (bottom) without batch normalization and dropout.

made as small as possible while still preserving fundamental features such as summing the input and output and different groups of residual blocks, as seen in Table 2. The hyperparameters chosen for training can be seen in Table 3. While truncation helped reduce overfitting, overfitting was still a major issue, as seen by the loss history in Figure 3.

- Use only two groups of residual blocks with 64 and 128 filters respectively.
- Each group contained two residual blocks.
- Due to the difference in size of the input images and the removal of two groups, the size before the fully connected layer is different.

3.3. Hyperparameter Optimization

For tuning our models we used the Hyperopt package [3]. This package allows for random hyperparameter search, as well as tree-structured parzen estimator (TPE). The TPE is a Bayesian approach to finding the best hyperparameter by modeling the probability of minimizing the loss given the previous trials [1]. The hyperparameters chosen for each model can be seen in Table 3.

3.4. Ensemble Model

After tuning our individual models, we implemented an ensemble model that combined all three of our multitask

[3]http://hyperopt.github.io/hyperopt

	AlexNet			GoogLeNet			ResNet		
	Subreddit	NSFW	Multi-task	Subreddit	NSFW	Multi-task	Subreddit	NSFW	Multi-task
Learning Rate	2.42e-3	3.36e-3	1.84e-3	4.25e-4	3.13e-4	3.77e-4	5e-4	1.75e-3	1.17e-3
Batch Size	100	100	100	64	64	64	64	64	64
Dropout	N/A	N/A	N/A	0.225	.351	0.033	N/A	N/A	N/A
Learning Rate Decay	0.994	0.975	0.971	0.948	0.972	0.985	0.91	0.989	0.90
Subreddit Weight	N/A	N/A	0.925	N/A	N/A	0.800	N/A	N/A	0.839

Table 3. Optimal hyperparameter values for all of our models, as found by HyperOpt

	AlexNet	GoogLeNet	ResNet
Standalone Subreddit	0.454	0.569	0.505
Standalone NSFW	0.476	0.837	0.782
Multitask Subreddit	0.455	0.639	0.498
Multitask NSFW	0.786	0.864	0.818

Table 4. F1 scores on our test set

models to see if we could improve further upon our best results. We tried out both a majority vote and weighted average ensembling scheme, and tested their performance on the test dataset. In addition, we also experimented with different weights for weighted average ensembling.

4. Results and Conclusions

4.1. Standalone Classification

For standalone subreddit classification GoogLeNet performed the best (58.4%), followed by ResNet (49.2%), and then AlexNet (46.3%). The F1 scores were also in the same order, with GoogLeNet performing the best (0.569), followed by ResNet (0.505), and then AlexNet (0.454).

For standalone NSFW classification GoogLeNet performed the best (95.3%), followed by ResNet (93.6%), and then AlexNet (90.9%). The F1 scores were also in the same order, with GoogLeNet performing the best (0.837), followed by ResNet (0.782), and then AlexNet (0.476).

4.2. Multitask Classification

Multitask classification gives a boost in performance for both subreddit classification and NSFW tagging in almost every case. AlexNet subreddit categorization test accuracy remains largely the same 46.3% for standalone versus 45.9% for multitask. AlexNet NSFW tagging test accuracy improved from 90.9% to 93.9%. This is a great example of how multitask learning can push a model in the right direction since for standalone NSFW tagging AlexNet predicted all SFW labels. The F1 score stayed roughly the same for subreddit classification (0.454 to 0.455), but was much better for NSFW tagging (0.476 to 0.786).

GoogLeNet subreddit categorization test accuracy improved from 58.4% to 64.4% and NSFW tagging test accuracy improved from 95.3% to 96.1%. The F1 score also increased from 0.569 to 0.639 for subreddit categorization and 0.837 to 0.864 for NSFW tagging.

ResNet subreddit categorization test accuracy improved from 49.2% to 51.9% and NSFW tagging test accuracy improved from 93.6% to 94.5%. The F1 score for subreddit categorization was about the same (0.505 to 0.498) while the F1 score for NSFW tagging increased from 0.782 to 0.818.

4.3. Ensembling

For majority-vote ensembling, we achieved an accuracy of 64.4% for subreddit classification and 96.1% for NSFW classification. For weighted-average ensembling, we applied a weight of 0.1, 0.2, and 0.7 to AlexNet, ResNet, and GoogLeNet, respectively (accounting for their relative performances). This led to an accuracy of 47.4% for subreddit classification and 94.9% for nsfw classifcation.

Thus, majority-vote ensembling does not lead to any change. On the other hand, we can clearly see that weighted average ensembling does not work well for our problem, as it performs significantly worse than our best performing model, GoogLeNet. It should be noted that changing the weights does not change this outcome.

We hypothesize that the reason weighted-ensembling performs poorly for our tasks is because our models are learning similar features, just to different degrees. Therefore, including AlexNet and ResNet in the ensemble model simply lowers down the accuracy from GoogLeNet, as they only contribute lower quality predictions. On the other hand, majority vote ensembling defaults to GoogLeNet when there is no clear majority, so its performance is also comparable.

4.4. Comparison

We would expect AlexNet to perform the worst in all cases, but the superior performance of GoogLeNet over ResNet is unexpected. This issue is attributed to truncation procedure. ResNet could only be limited to the size it currently is in order to retain key features. However, these key features were built for very deep networks unlike this more shallow version. GoogLeNet on the other hand can be limited to a single inception layer, allowing for more truncation and less overfitting.

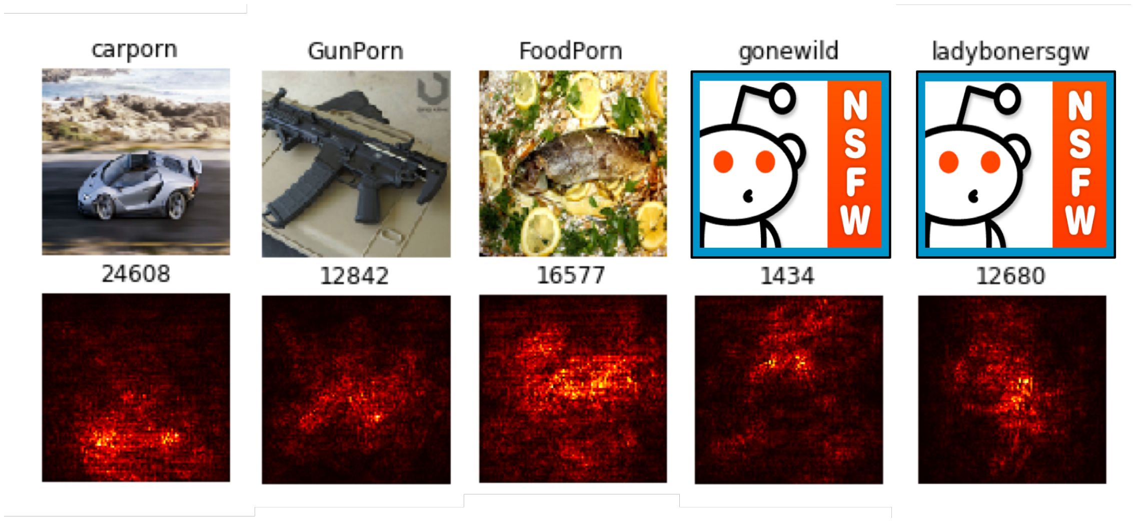

Figure 4. Saliency maps for examples from five select subreddit communities. The brighter the pixel, the more that particular neuron is activated. The patches of red pixels represent the areas in the image that most influence the classification result of the image.

		AlexNet		GoogLeNet		ResNet	
		Standalone	Multi-task	Standalone	Multi-task	Standalone	Multi-task
Train	Subreddit	59.0	63.3	80.5	93.6	76.4	79.8
	NSFW	91.5	96.2	98.9	98.7	96.0	98.1
Val	Subreddit	46.6	46.0	58.7	64.1	51.3	54.4
	NSFW	92.0	95.1	95.8	96.7	94.5	95.7
Test	Subreddit	46.3	45.9	58.4	64.4	49.2	51.9
	NSFW	90.9	93.9	95.3	96.1	93.6	94.5

Table 5. Accuracy results (in %) of our 3 architectures trained on each of the tasks for the train, val, and test datasets

4.5. Human Performance

In order to characterize how close our best models compare to humans, we performed hand labeling of reddit images. We familiarized ourselves with the 20 different subreddit communities and some example NSFW images, and proceeded to label 150 randomly sampled images. We achieve an average subreddit labeling accuracy of 84%, and average NSFW labeling accuracy of 98%. This demonstrates that there is room for improvement for our current categorization models.

4.6. Class Confusion

The best performing model is GoogLeNet with multi-task learning. We investigate this model in more detail first looking at the confusion matrix in Figure 5. The category for r/FoodPorn has the highest accuracy. This is likely due to the fact that food shares less characteristics with the other categories, and that food likely does not occur too often in other categories. The worst accuracy is for r/creepy. Creepiness is a vague concept and our neural network architecture does not seem to grasp what creepy means. We see a confounding of categories that seem to make sense logically. Subreddits r/ruralporn and r/EarthPorn are confused because they both contain pictures of nature. Subreddits r/dogpictures and r/MilitaryPorn are confused likely due to the fact that military dogs get lots of upvotes. Subreddits r/ArchitecturePorn and r/CityPorn are confused because cities contain many great pieces of architecture.

4.7. Saliency Maps

Next we look at example saliency maps to see if our model highlights parts of images that contain features we think are characteristic of its subreddit class. Looking at Figure 4, we see 5 example saliency maps from 5 of our 20 subreddit classes.

We can see from the r/carporn saliency map that the algorithm thinks the wheels are important for the car categorization, lighting up the pixels that correspond to the wheels.

In the r/GunPorn saliency map we see the magazine and handle of the gun light up and a dimmer outline of the entire shape of the gun.

In the r/FoodPorn map we see that the pixels which correspond to the fish light up.

The subreddits that contain NSFW images are redacted for the sake of this report, but we still show their corre-

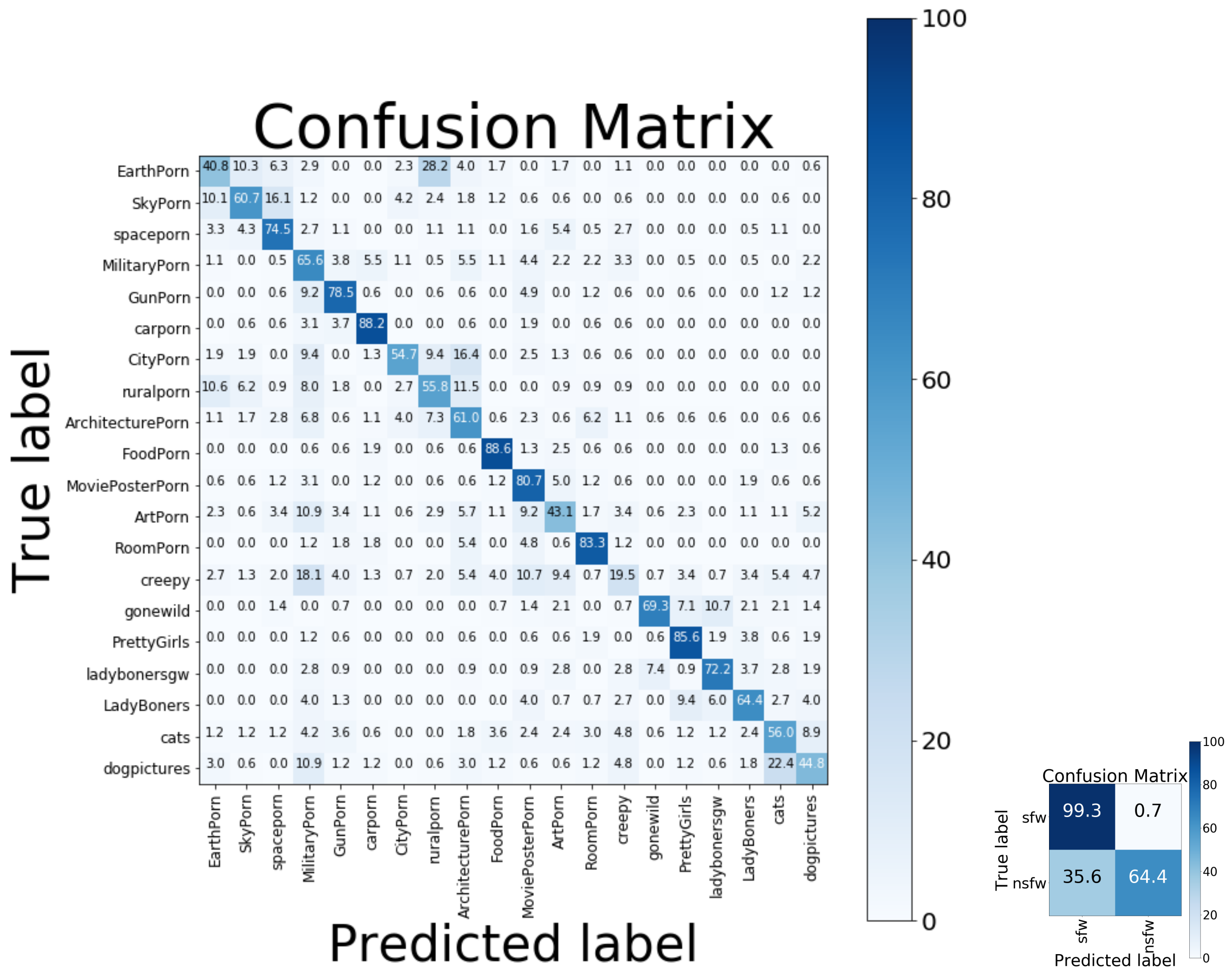

Figure 5. Confusion matrix for subreddit categorization (left) and NSFW tagging (right) from training on GoogLeNet using multitask learning.

sponding saliency maps. The example from r/gonewild is of a naked woman on her hands and knees facing away from the camera. In the saliency map you can clearly the pixels corresponding to her back, buttocks and legs light up.

The example from r/ladybonersgw contains a naked man with his hands on his hips, facing to the right in a state of arousal. In the saliency map, the pixels corresponding to his torso, arms, and penis light up. It is also interesting to note that the brightest part of the saliency map corresponds to the man's camera phone. People who post naked pictures of themselves most often do so by taking selfies. Our algorithm may be picking up camera phones in NSFW selfies as a distinguishing feature.

4.8. Future Work

All models that we have implemented currently overfit; further experimenting with truncated architectures or introducing more regularization would be valuable to investigate in order to reduce overfitting of the training data set. Additionally, many of the data augmentation methods suggested in [6] have been shown to greatly decrease overfitting – these are worth trying as well.

After further optimization of NSFW tagging predictions, it would be interesting to investigate if image visualization of NSFW labeled images could be used to auto-blur the NSFW aspects of the image. For instance, one may wish to see the posts, but not be exposed to adult content. In this case, an auto-bluring algorithm would prove to be very useful.

References

[1] J. Bergstra, D. Yamins, and D. Cox. Making a science of model search: Hyperparameter optimization in hundreds of dimensions for vision architectures. *Proceedings of Machine Learning Research*, 28, 2013.

[2] R. Caruana. Multitask learning. In *Learning to learn*, pages 95–133. Springer, 1998.

[3] T. G. Dietterich. Ensemble methods in machine learning. In *International workshop on multiple classifier systems*, pages 1–15. Springer, 2000.

[4] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 770–778, 2016.

[5] S. Ioffe and C. Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. *arXiv preprint arXiv:1502.03167*, 2015.

[6] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In F. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 25*, pages 1097–1105. Curran Associates, Inc., 2012.

[7] S. Ruder. An overview of multi-task learning in deep neural networks. `http://sebastianruder.com/multi-task/`, 2017.

[8] K. Simonyan, A. Vedaldi, and A. Zisserman. Deep inside convolutional networks: Visualising image classification models and saliency maps. *CoRR*, abs/1312.6034, 2013.

[9] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich. Going deeper with convolutions. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–9, 2015.