# Judging an academic paper by its cover

Project 17

**Marcus Jonsson Ewerbring**[*] **Marcus Österberg**[†]
**Jonas Valfridsson** [‡] **Love Almgren**[§] **Cornelis Strohkirch**[¶]

December 17, 2020

**Abstract**

When submitting a paper to a conference the paper will go through peer-reviews and evaluations in order to determine whether it should be either accepted or rejected. In this paper it was investigated if a machine learning method could find visual features or biases that increases the chance of getting accepted. Previous work from Huang, Jia-Bin [2] was evaluated with a new model based on Resnet32 and Resnet18. The new model was applied on a dataset containing accepted and rejected papers from the conferences ICLR and MIDL. The results indicate that the dataset from Huang, Jia-Bin could include data leakage. No visual biases between the accepted and rejected papers of ICLR and MIDL were found.

[*]marcusew@kth.se
[†]maoste@kth.se
[‡]jonas@valfridsson.net
[§]loveal@kth.se
[¶]strohk@kth.se

# Contents

# 1 Introduction

The amount of conferences papers being submitted has seen a noteworthy increase in the past few years. For example, in 2018 there was 935 paper submissions to the International Conference of Learning Representation (ICLR) and in 2020 there was 2594 papers, an increase of 177%[7, 15, 8, 16] Each conference paper needs to be peer reviewed, which today is done by humans. This human process is subject to various problems ranging from personal bias to consistency issues.

This paper attempts to uncover if there exist visual biases on the ICLR conference papers by using convolutional neural networks (CNN's) to predict if a paper gets accepted or not. The trained model is analyzed in order to understand what the network has learned to conclude if visual biases are present or not. A visual bias is defined as anything in the visual layout of the paper that can be overused by writers solely to achieve a better chance to get the paper accepted. For example, if an excessive amounts of images, impressive mathematical equations or an extensive references section increase the chance of getting the paper accepted, it would be considered a visual bias.

The network was trained on papers from the conferences MIDL and ICLR, and achieves an accuracy of 68% after removing information leakage from the papers.

An attempt to recreate the results from Huang, Jia-Bin [2] was also done, and achieved comparable results in regards to the accuracy presented by the authors. However, augmenting the dataset, it was discovered that the dataset could contain data leakage.

The purpose of this network is not to completely automate the peer reviewing process of papers. The network, if functional, should act as an aiding tool for both writers and reviewers to give a hint on how the visual layout of the report will affect the chance of getting the paper accepted or not.

## 1.1 Ethics and Sustainability

If this model could detect visual bias or suggest a visual layout that increases the chance of the paper getting accepted in to a conference, it could have ethical consequences. This could result in that the user that are aware of this format get an unfair advantage against other people that submit papers. However, if it detects a visual bias it could also have the opposite consequence where the conference have to re-evaluate the criteria for getting accepted and promote scientific content.

If the data set contains a racial/ethnic bias, that is to say that the data shows that one group of people generally gets rejected or accepted more often, the model that is built will also contain this bias. Therefore, unless we account for these biases when the model is built, the model might exacerbate racial/ethnic biases that already exist. Which could potentially work against the sustainable development goal "reduced inequalities" [19].

If we could replace some part of the human peer-review process and filter out papers, less people would need to spend time on peer-reviewing papers. This might lower the number of human jobs in the area - which might impact the social sustainability of the project negatively, seeing as the process would be removing jobs from humans. However, peer reviewers are generally not (directly) paid for their work and therefore we are not removing paying jobs. This might either result in that less researchers in general are

needed or that researchers might have more time to do other more productive jobs. This could either benefit or be to the detriment of goal 8 "promote sustained economic growth" [18]. The environmental sustainability impact of the process would be the power consumption necessary to run the network.

## 1.2 Related work

The related work section is divided into two sections, since there are primarily two approaches to the problem that we have found, either a visual- or non-visual approach.

### 1.2.1 Related work with a visual approach

Huang, Jia-Bin in "Paper Gesalt" [2] presents a solution to uncover visual biases in a conference paper. The authors used a dataset consisting of CVPR (2013-2018) and ICCV (2013-2017). The task was to distinguish between good (accepted) and bad (rejected) papers, and they used the conference papers as accepted papers and the workshop papers as rejected papers. An accuracy of 92% was achieved when the model was trained on papers between 2013-2017 and CVPR 2018 was used as a test-dataset. They found that too little content (less than 8 pages) or the absence of illustrative figures in the first two pages were some visual indicators of bad papers, while colorful illustrations and impressive math equations were visual indicators of good papers. The authors noticed that a model trained on their dataset is not applicable on any other conferences, because there is a visual difference between conferences. The authors suggest a future study were a dataset is created from ICLR, where both accepted and rejected papers are given by the conference.

### 1.2.2 Related work with a non-visual approach

Jet et al in "Predicting Conference Paper Acceptance" [3], tries to predict acceptance rate in the conference ICLR 2017. The network used several features as input, for example, title length, number of authors, number of references and more. There was a total of 18 features, were all were integer numbers, booleans or floating numbers. They found that a SVM with the RBF kernel performed the best with an accuracy of 71%.

M. Skorikov and S. Momen in "Machine learning approach to predicting the acceptance of academic papers" [14] achieved an impressive accuracy of 81% using Random Forest classifier. The authors used the PeerRed dataset which contains 14 700 papers, where they used the attributes title length, number of tables, number of figures, number of citations and length of the literature review as features.

## 1.3 Problem description

Previous works [2] have tried to identify visual biases in CVPR and ICCV, in this work we try to do the same for ICLR and MIDL.

Given the visual appearance of a paper it is attempted to classify if it will be accepted or rejected. The result is then interpret by analyzing what the classifier learned to see if

it has learned something that could be seen as a visual bias.

The following are the research questions addressed:

- Can we reproduce the results from *Deep Paper Gestalt*?
- How well can visual appearance predict acceptance to ICLR and MIDL?
- Are there any visual biases in ICLR or MIDL?

# 2 Method

In this section we present how we created the data-set, picked the model and conducted the experiments to answer our research questions.

## 2.1 Data representation

*Deep Paper Gestalt* [2] represented papers as 2x4 blocks of images and we chose to implement the same approach. We also tried to represent the whole paper as a single block where each page is represented by a channel. More details of these representation are given in the following two sections.

These two methods are explained in the following subsections. Each page in the dataset was scaled to the resolution 256x256. Each dataset were in RGB format instead of grey-scale in order to keep potential features that were in the red, green or blue range.

### 2.1.1 Array of RGB pages

This data-representation was inspired by the paper *Deep Paper Gestalt* from Huang, et al [2]. They arranged 8 pages from each paper in 2 rows of 4 images each to capture the information in a conference paper, see figure 1.

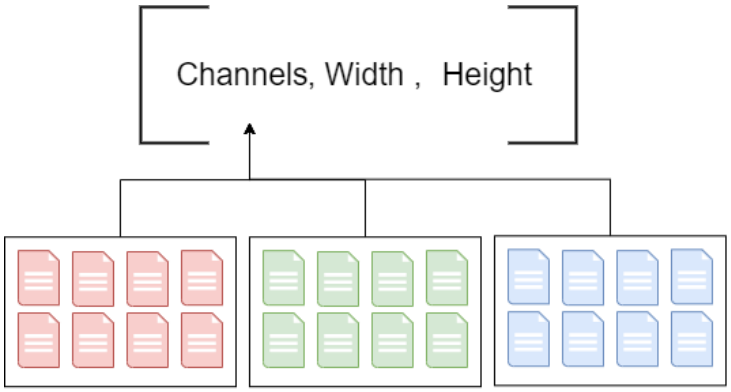



**Figure 1:** Illustrates the arrangement of 8 pages from a paper into one image

The width of the data sample is decided by the width from each paper. The data sample will have the format $[channels, width, height]$, where each channel contains a red, blue and green value.

### 2.1.2 24 channel input

The *24 channel input* dataset representation arrange the first 8 pages of the report as a separate channel in the datapoint, see figure 2.

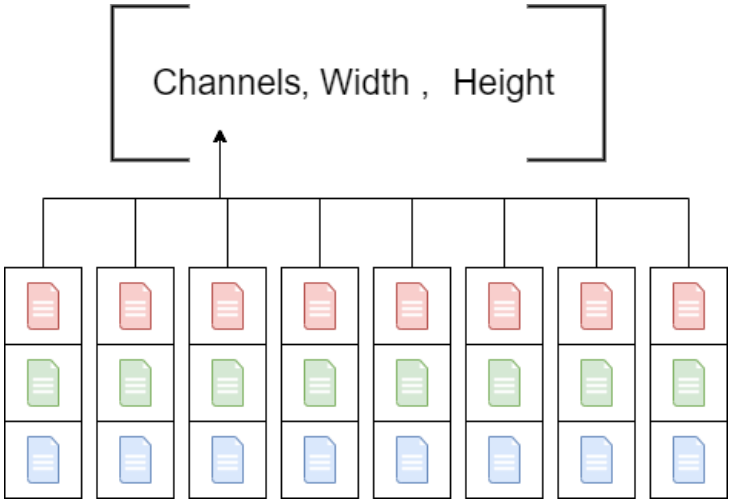



**Figure 2:** Illustrates the storage of pages into a datapoint.

In figure 2 each datapoint will have the format $[channel, width, height]$, were each channel is a page in the paper. Due to that a data point consists of 8 pages which have 3 channels each, a total of 24 channels per datapoint is needed.

## 2.2 Models

In this paper we use two different variants of Resnet, Resnet18 and Resnet34, to conduct our experiments.

### 2.2.1 Resnet18 & Resnet34

Our implementation used both Resnet18 and Resnet34. Resnet18 is a smaller network containing approximately 11 million parameters, whereas Resnet34 is contains 21 million parameters [11]. They were chosen as both as *Deep Paper Gestalt* used Resnet18 (and tested using Resnet35/Resnet50)[2] and as Resnets have good performance at challenging computer vision tasks [17]. We used resnet34 to be able to model a more difficult problem given that it has more parameters then resnet18.

Both of these networks were modified to accommodate for our number of classes and input sizes. The network was modified to only have a single output node with dropout and a probability of 0.5, as our classification only requires true or false, for rejected or accepted. When using the 24-channel input version of the dataset, the first convolutional layer of the networks was modified to be able to accept a 24-channel input. We used a *cosine annealing* [5] scheduler in our implementation in order to perform "warm restart" of the learning rate which could avoid getting local minimas. In preliminary tests we noticed that cosine performed in general better than a step based scheduler. The implementations used pre-trained versions of network, with an Adam optimizer.

### 2.2.2 Implementation details

According to the results from the related work section, see 1.2, there has not been much research on predicting if a paper will get accepted to a conference based on the appearance of the paper. Because of that we don't know what the optimal parameters for a network is. A hyperparameter search was performed in order to find the parameters with the highest performance. The investigated parameters were the following, where $\eta$ is the learning rate, $d$ is the weight decay and $B$ is batch size:

**Table 1:** The values tested for the coarse-grained search

| | | | | |
|---|---|---|---|---|
| $\eta$ | 1e-4 | 5e-5 | 2.5e-5 | 1e-5 |
| $d$ | 1e-6 | 1e-7 | 1e-8 | N/A |
| $B$ | 10 | 25 | 50 | N/A |

**Table 2:** The values tested for the fine-grained search

| | | | | |
|---|---|---|---|---|
| $\eta$ | 1e-5 | 5e-5 | 3e-5 | 7e-4 |

The hyperparameter search was done on both the dataset square of RGB and 8 channel input. Each network used a pretrained network implementation as an initialization of the network and was trained for **10** epochs. This experiment was repeated for both resnet18 and resnet32.

The main idea for the search was to first perform a coarse-grain search for each parameter, where several values from a relatively large range were tested. Once the best value for that range was found, a more fine-grained search was performed i.e on a smaller range of values, close to the best value found.

Another important part of the hyperparameter search was to find the number of epochs to train the network before overfitting occurs. This was done by carefully investigating the charts.

The hyperparameters which achieved the highest performance are presented bellow:

- Learning rate: 0.00005
- Weight decay: 1e-7
- Batch size: 10
- Over-fitting occurs after 7 epochs

It was noticed that after 7 epochs the model started to overfit on the ICLR  MIDL dataset and thus we limited the tests conducted on ICLR & MIDL to 7 epochs of training (which corresponds to approximately 1000 update steps).

### 2.2.3 Data Augmentations

Data augmentation is a technique used in deep learning to reduce overfitting of a model. Generally deep learning models tend to overfit when the dataset does not contain enough samples. This problem can be addressed with various techniques, one of them is data

augmentation [13]. In the case of images, it is called image augmentation and our implementation used noise injection, vertical flipping, kernel filters and color space transformations.

- Noise injection adds random Gaussian noise to the image.
- Vertical flipping randomly flips the image around the vertical axis, creating mirrored images.
- Kernel filters add random blur to the images, using a Gaussian blur filter.
- Color space transformations was implemented to change the hue of the image.

Our implementation applied either one or no augmentations to each training sample. Each augmentation had a 16 percent chance of being chosen, resulting in 64 percent of the papers having some kind of augmentation and 36 percent of the papers having no augmentations. Augmentations were applied to the training set, they were not applied to the validation or test sets.

## 2.3 Experiments

In this section the experiments performed on the model and how data was collected are presented. This section contains the following subsections *Selecting conference papers*, *Web scraping script*, *Data leakage* and *Evaluation* .

## 2.4 Selecting conference papers

All conference papers were scraped from OpenReview [9]. It was scraped papers from ICLR (2018-2020) and MIDL (2018-2020). In total 4825 papers were downloaded of which 1724 papers were accepted. These conferences were used because they provided PDF's for both accepted and rejected papers.

| Conference | # Accepted | # Rejected | Acceptance percentage |
|---|---|---|---|
| MIDL 2020 | 106 | 28 | 79.1% |
| MIDL 2019 | 47 | 10 | 82.4% |
| MIDL 2018 | 47 | 36 | 56.6% |
| ICLR 2019 | 502 | 911 | 35.5% |
| ICLR 2018 | 335 | 574 | 36.8% |
| ICLR 2020 | 687 | 1522 | 31.1% |
| Total | 1724 | 3081 | 35.9% |

**Table 3:** Dataset paper statistics

It is worth noting that we have less papers in the dataset then available on openreview. This is due that some of them were incompatible with the system converting the data to the used format of our system.

The data was divided in a 3-way split where ICLR-2020 and MIDL-2020 were the testdata. Papers from 2018 and 2019 were used as traindata were 70% was used for training and 30% for validation.

### 2.4.1 Web scraping script

The OpenReview API [10] to build a scraper using python 3.8. The scrapers code can be located in our Github repository [12]. Crude usage documentation can be found in the README of that repository. The script produces a CSV file with the fields id, authors, abstract, conference, year, title, accepted and image_path as well as a directory containing the PDF of each paper. The image_path field can be used to map rows in the CSV to papers.

### 2.4.2 Data leakage

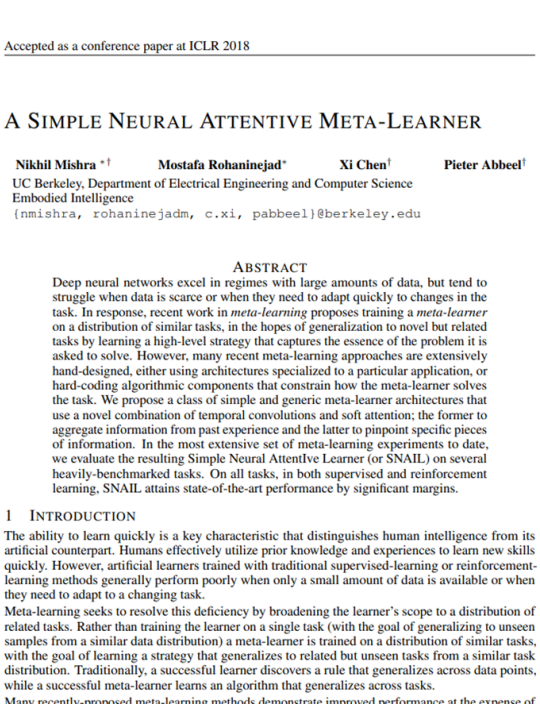
Papers scraped from OpenReview.net [9] contained visual information that gave away information about the target, i.e if it was accepted or not. In the *header* of each page it was either written "Published in ICLR XXXX" or "Under review" which tells the network if the paper was accepted or not. A more subtle feature was that for each paper, if it got accepted: the authors were present on the front page. If it was not accepted the authors were anonymous. This shows that our dataset could include data leakage and the impact of this will be investigated in 2.3.

### 2.4.3 Data leakage experiment

In order to investigate the effect of data leakage the network was trained with the best found hyper-parameters from section 2.2.2, with the following modification of the datasets.

- Remove front page, remove header.
- Remove front page, keep header.
- Keep front page, remove header.
- Keep front page, keep header.

As mentioned in 2.4.2 the header and front page contains information that could affect the performance. The experiments were designed to find if keeping either the header or front page would result in data leakage, see figure 3 and 4. This test was repeated for **both** the datasets *square of RGB pages* and *24 channel input*.



Accepted as a conference paper at ICLR 2018

A SIMPLE NEURAL ATTENTIVE META-LEARNER

**Nikhil Mishra** *† **Mostafa Rohaninejad*** **Xi Chen**† **Pieter Abbeel**†
UC Berkeley, Department of Electrical Engineering and Computer Science
Embodied Intelligence
{nmishra, rohaninejadm, c.xi, pabbeel}@berkeley.edu

ABSTRACT

Deep neural networks excel in regimes with large amounts of data, but tend to struggle when data is scarce or when they need to adapt quickly to changes in the task. In response, recent work in *meta-learning* proposes training a *meta-learner* on a distribution of similar tasks, in the hopes of generalization to novel but related tasks by learning a high-level strategy that captures the essence of the problem it is asked to solve. However, many recent meta-learning approaches are extensively hand-designed, either using architectures specialized to a particular application, or hard-coding algorithmic components that constrain how the meta-learner solves the task. We propose a class of simple and generic meta-learner architectures that use a novel combination of temporal convolutions and soft attention; the former to aggregate information from past experience and the latter to pinpoint specific pieces of information. In the most extensive set of meta-learning experiments to date, we evaluate the resulting Simple Neural AttentIve Learner (or SNAIL) on several heavily-benchmarked tasks. On all tasks, in both supervised and reinforcement learning, SNAIL attains state-of-the-art performance by significant margins.

1 INTRODUCTION

The ability to learn quickly is a key characteristic that distinguishes human intelligence from its artificial counterpart. Humans effectively utilize prior knowledge and experiences to learn new skills quickly. However, artificial learners trained with traditional supervised-learning or reinforcement-learning methods generally perform poorly when only a small amount of data is available or when they need to adapt to a changing task.

Meta-learning seeks to resolve this deficiency by broadening the learner's scope to a distribution of related tasks. Rather than training the learner on a single task (with the goal of generalizing to unseen samples from a similar data distribution) a meta-learner is trained on a distribution of similar tasks, with the goal of learning a strategy that generalizes to related but unseen tasks from a similar task distribution. Traditionally, a successful learner discovers a rule that generalizes across data points, while a successful meta-learner learns an algorithm that generalizes across tasks.

Many recently-proposed meta-learning methods demonstrate improved performance at the expense of

**Figure 3:** Illustrates an example of an accepted paper, **note** that the header reveals that the paper is accepted and that the authors are available. The accepted paper is from Mishra, et al [6].

Under review as a conference paper at ICLR 2018

TRAINING AUTOENCODERS BY ALTERNATING MINIMIZATION

**Anonymous authors**
Paper under double-blind review

ABSTRACT

We present DANTE, a novel method for training neural networks, in particular autoencoders, using the alternating minimization principle. DANTE provides a distinct perspective in lieu of traditional gradient-based backpropagation techniques commonly used to train deep networks. It utilizes an adaptation of quasi-convex optimization techniques to cast autoencoder training as a bi-quasi-convex optimization problem. We show that for autoencoder configurations with both differentiable (e.g. sigmoid) and non-differentiable (e.g. ReLU) activation functions, we can perform the alternations very effectively. DANTE effortlessly extends to networks with multiple hidden layers and varying network configurations. In experiments on standard datasets, autoencoders trained using the proposed method were found to be very promising and competitive to traditional backpropagation techniques, both in terms of quality of solution, as well as training speed.

1 INTRODUCTION

For much of the recent march of deep learning, gradient-based backpropagation methods, e.g. Stochastic Gradient Descent (SGD) and its variants, have been the mainstay of practitioners. The use of these methods, especially on vast amounts of data, has led to unprecedented progress in several areas of artificial intelligence. On one hand, the intense focus on these techniques has led to an intimate understanding of hardware requirements and code optimizations needed to execute these routines on large datasets in a scalable manner. Today, myriad off-the-shelf and highly optimized packages exist that can churn reasonably large datasets on GPU architectures with relatively mild human involvement and little bootstrap effort.

However, this surge of success of backpropagation-based methods in recent years has somewhat overshadowed the need to continue to look for options beyond backprogagation to train deep networks. Despite several advancements in deep learning with respect to novel architectures such as encoder-decoder networks and generative adversarial models, the reliance on backpropagation methods remains. While reinforcement learning methods are becoming increasingly popular, their scope is limited to a particular family of settings such as agent-based systems or reward-based learning. Recent efforts have studied the limitations of SGD-based backpropagation, including parallelization of SGD-

**Figure 4:** Illustrates an example of an rejected paper, **note** that the header reveals that the paper is rejected and that the authors anonymous. The rejected paper is from kudugunta, et al [4].

### 2.4.4 Deep Paper Gestalt experiment

The authors of *Deep Paper Gestalt* [2] reached an accuracy of 92% on the test dataset. We conducted experiments to test if the deep gestalt dataset contains data-leakage that were not removed by the authors. The model was trained with hyper-parameters found in 2.2.2, but with 50 epochs as that was what was used in *Deep Paper Gestalt*.

First,the result of the model should give comparable results on the *Deep Paper Gestalt* dataset as provided by the authors. Second, different parts of the papers were removed in their dataset starting with the page-numbers then the last two pages to show the impact on the performance, or potential data leakage. Third, removing random squares across the papers and also randomly picked pages in an attempt to show that page numbers and the last two papers were major contributors to the impressive numbers.

The page-number was investigated as a leakage because it differed between conferences and workshop papers, see table 4.

**Table 4:** Illustrates the Max Page Number Conference Vs Workshops. The Max Page Number is the highest page number a conference or workshop will end at.

| | Workshop | Conference |
|---|---|---|
| 2016 CVPR | 198 | 6043 |
| 2015 CVPR | 160 | 5565 |
| 2017 CVPR | 179 | 7483 |
| 2018 CVPR | 2628 | 9474 |
| 2015 ICCV | 171 | 4713 |

## 2.5 Evaluation

For the training, validation and test dataset the accuracy was calculated and the *Area Under the ROC Curve* (AUC score)[1] to get a sense of how the models were performing. The AUC score represents the area under a curve formed by the true positive rate plotted against the true negative rate. The AUC score is a value between 0 and 100%, which represents how well the model can separate positive and negative samples.

Class activation maps (CAM) were used to visually inspect what features the models learned. This was the main tool for identifying visual biases that the model learned. Due to the implementation of CAM it is not applicable on the 24-channel dataset.

# 3 Result

In this section the results from the experiments in section 2.3 are presented.

## 3.1 ICCV & CVPR

In this section the results from the reports deep paper gestalt experiment which was presented in section 2.4.4 are shown. The network was first trained in **50 epochs** in order to replicate the results from *Deep Paper Gestalt*, the network got a test accuracy of 91.2%. In table 5 the network is trained in **7 epochs**, to make it comparable with the results from section 3.2.1.

It is worth noting that if the network only would answer accept on **all data points** if would get an accuracy of **79%**. In the following table we refer to page number as (PN), last two pages as L2P front page as FP, random page as RP and random boxes as RB.

**Table 5:** Shows the results from the Deep Paper gestalt experiments

| | R32 AUC | R32 Accuracy | R18 AUC | R18 Accuracy |
|---|---|---|---|---|
| Original | 93.0% | 89.2% | 93.9% | 90.2% |
| No PN | 88.3% | 88.1% | 89.5% | 90.7% |
| No L2P | 91.6% | 87.0% | 90.9% | 87.1% |
| Drop two RP | 92.7% | 89.7% | 94.4% | 89% |
| Drop RB | 93.4% | 89.4% | 93.9% | 90.3% |
| No (PN or L2P) | 86.8% | 85.7% | 86.2% | 85.1% |
| No (PN, L2P or FP) | 83.1% | 83.7% | 84.2% | 84.1% |

### 3.1.1 CAMs, ICCV & CVPR

The following figures shows cams from the experiments presented in the table above.

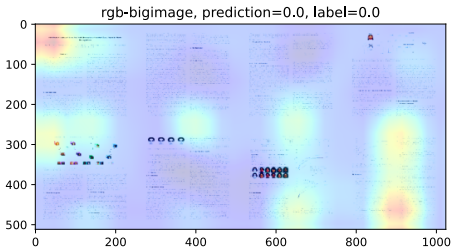



**Figure 5:** CAM generated with original deep gestalt dataset. Accuracy Resnet32 89.2%

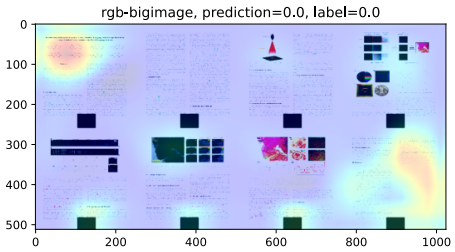



**Figure 6:** CAM generated without page numbers. Accuracy Resnet32 88.1%

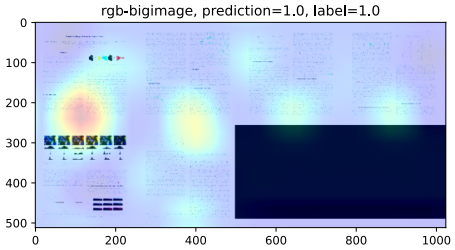



**Figure 7:** CAM generated without last two pages. Accuracy Resnet32 87%

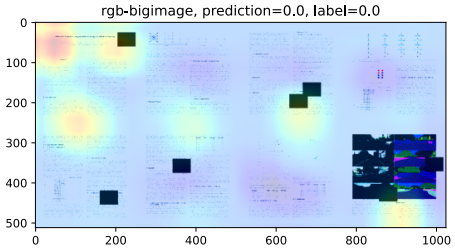



**Figure 8:** CAM generated without random patches. Accuracy Resnet32 89.4%

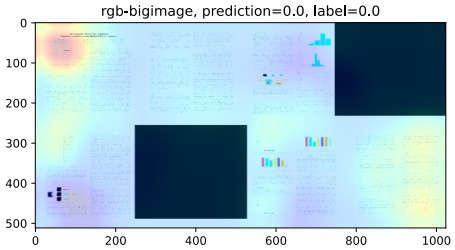



**Figure 9:** CAM generated where two random pages were dropped. Accuracy Resnet32 89.7%

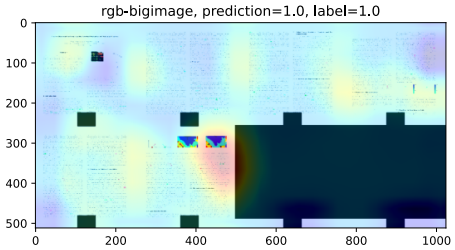



**Figure 10:** CAM generated without last two pages and page numbers. Accuracy Resnet32 85.7%

## 3.2 ICLR & MIDL

The results from data leakage experiment are presented in the following section. The accuracy for the test set and AUC score are presented in table 6 for square of RGB pages and table 7 for 24-channels. Figures of loss curves can be found in section 6 appendix.

It is worth noting that if the network only would answer accept on **all data points** it would get an accuracy of **66.1%** on the ICLR dataset.

**Table 6:** Test accuracy and AUC score for square of RGB pages dataset. R32 stands for Resnet32 and R18 stands for Resnet18. *L* stands for that the data contains leakage while *NL* stands for that the data do not contains leakage.

| | R32 AUC | R32 Accuracy | R18 AUC | R18 Accuracy |
|---|---|---|---|---|
| Original *L* | 96,5% | 97,5% | 99.7% | 97.1% |
| No front page *L* | 99,3% | 94,1% | 98.4% | 95.5% |
| No header *L* | 95,2% | 97,2% | 95,9% | 97% |
| No front page and header *NL* | 71,7% | 68,7% | 72,3% | 69,1% |

**Table 7:** Test accuracy and AUC score for 24-channels dataset. R32 stands for Resnet32 and R18 stands for Resnet18

| | R32 AUC | R32 Accuracy | R18 AUC | R18 Accuracy |
|---|---|---|---|---|
| Original | 94.8% | 96.0% | 94.0% | 95.6% |
| No front page | 94.6% | 96.0% | 95.7% | 96.6% |
| No header | 90.4% | 92.9% | 90.0% | 91.8% |
| No front page and header | 56.6% | 61.2% | 59.6% | 63.3% |

### 3.2.1 CAMs, ICLR & MIDL

Figure 11 shows a CAM generated with our dataset when images include front page and headers, while Figure 14 shows a CAM generated when neither of these are included.

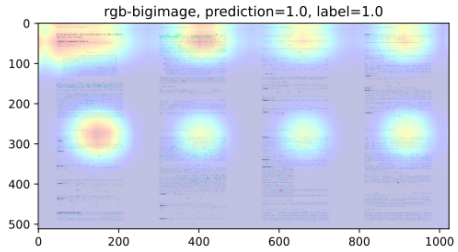



**Figure 11:** CAM generated with front page and headers. Accuracy Resnet32 97.5%

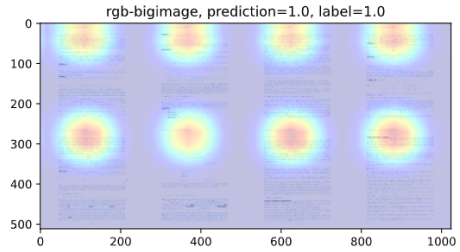



**Figure 12:** CAM generated without front page and headers. Accuracy Resnet32 94.1%

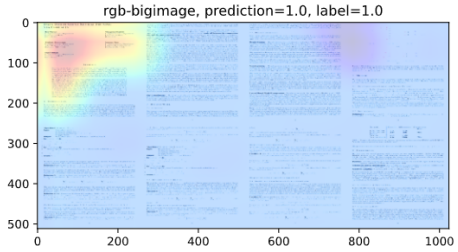



**Figure 13:** CAM generated without header but with front page. Accuracy Resnet32 97,2%

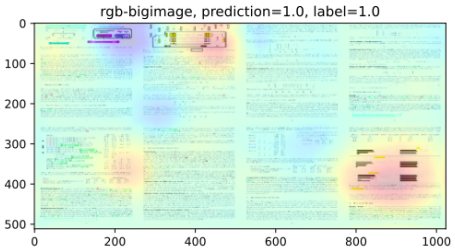



**Figure 14:** CAM generated without front page and headers. Accuracy Resnet32 68.7%

# 4 Discussion

As we can see in our results, section 3.1, our accuracy on the test set when using the *Deep Paper Gestalt* dataset, 91.2%, is comparable to the original *Deep Paper Gestalt* result where they achieved 92% accuracy [2] on the same test set. This shows that our network implementation could reproduce the results of the original network that was used in *Deep Paper Gestalt*. We also used an implementation of Resnet18, however there are differences in our implementation, i.e. we use a single output node whereas the other network uses two.

The dataset used in *Deep Paper Gestalt* treats workshop papers as "bad" and conference papers as "good". This could differ from the original task of classifying good versus bad papers - the workshop papers are still accepted to the workshops, which might suggest that they are good papers. When training a network using this dataset the network more specifically learns to differentiate between conference papers and workshop papers. This is not necessarily in-line with the task of discerning between accept and reject.

To test the dataset used in *Deep Paper Gestalt*, we trained networks when removing parts which we thought could be data leakage - we had for example seen that general workshop papers had lower page numbers than conference papers, see table 4 .

In table 4 we can see that page numbers is a data leakage for most CVPR conferences. If a paper has a page number larger than 200 it is guaranteed to be accepted, this follows from a difference between how the conference and workshops does the page enumeration. This leakage can also be observed in the CAM images in section 3.1, for example figure 6 shows that the model focuses around the page number area to make its prediction. We can also observe a significant drop of performance when we remove the page numbers. When running experiments where we removed random boxes from the papers, the accuracy did not decrease. This could indicate that the specific boxes that contains the page numbers are the ones relevant for the performance drop and not the fact that we removed arbitrary information.

A second source of potential leakage that we discovered were the last two pages - the authors of *Deep Paper Gestalt* mention that the last two pages were of importance for their classifier so they were aware of this. Anecdotally we have observed that rejected papers tend to be shorter and therefore have more empty space towards the end, we believe this has to do with it being a workshop. It is not obvious whether or not this is a feature or leakage, we believe it is leakage due to our observation that the workshop papers are shorter.

A third source of potential leakage, or bias, seem to be the author field. From observing the CAMs we believe that the model often uses the author field to determine if a paper was rejected. It is up for debate whether this is an actual feature or leakage. We believe if the purpose is to find visual biases then the author field should not be relevant therefore it should be considered leakage if you're hoping for the model to learn visual biases. The author field is clearly a type of bias though, especially if it is visible to the reviewers during the review process.

However leakage does not account for all of the performance of the network on ICCV and CVPR so there seems to be some visual biases present. From studying the CAMs it does appear that an abundance of images and impressive text sections does have an impact on the acceptance rate of the paper as also shown by the authors of *Deep Paper Gestalt*.

It is worth noting though that although leakage did not account for all of the performance on our test set - on our validation set when the front page, numbers and last two pages were removed the accuracy was 79% which is equivalent to just guessing that each paper is rejected. For the validation dataset our model seemed to have learned nothing once the 'leakage' was removed.

### 4.1 Results for MIDL and ICLR

For array of RGB images dataset, the results in Table 6 show that the Resnet18 model, when removing both front pages and headers, achieved an accuracy of 69.1% with an AUC-score of 72.3%. The Resnet32 model with the same removed features achieved an accuracy of 68.7% and AUC-score of 71.7%. As a reference, guessing only rejected on the test-set achieves an accuracy of 66.2%, our Resnet18 model performs 2.9% better while the Resnet32 model performs 2.5% better. Judging by this, our networks did not learn to differentiate between accepted and rejected papers. This could point towards that there is no visual bias to be found in the ICLR & MIDL dataset.

We can also see that the network could learn to judge the papers using the headers which either contained the authors name or "authors name withheld" as well as the front page which contained the same information. This can be seen both in the drop of accuracy and AUC-score in Table 6 as well as the in the generated CAMs in Figures 11-14.

The results of the 24 channel input dataset performed on average worse, but the same tendency to learn to look in the page header or front page was seen.

It is worth noting that MIDL, ICLR have a different format than ICCV and CVPR. ICCV and CVPR have 2 columns of text per page while MIDL and ICLR only have 1 column per page. The authors from *Deep Paper gestalt* pointed out that a trained model on ICCV and CVPR would not perform well on other papers. This could indicate that the format of the paper influence the behavior of the network.

## 5 Conclusion

In this report a reproduction of *Deep Paper Gestalt* was performed with a modified version of Resnet18 and Resnet32. A new custom dataset containing papers from ICLR and MIDL was created and evaluated with the model.

The results could indicate that there is visual bias present when trying to differentiate between conference and workshop papers. No visual biases were discovered for rejected versus accepted papers from ICLR and MIDL.

Future work could be to increase the dataset with other conferences with similar formatting style or to use different models to train with the dataset.

# References


[1] Google. *Classification: ROC Curve and AUC*. URL: https://developers.google.com/machine-learning/crash-course/classification/roc-and-auc.

[2] Jia-Bin Huang. "Deep Paper Gestalt". eng. In: (2018).

[3] William Jen, Shichang Zhang, and Muyun Chen. "Predicting Conference Paper Acceptance". In: ().

[4] Sneha Kudugunta et al. *Training Autoencoders by Alternating Minimization*. 2018. URL: https://openreview.net/forum?id=B1D6ty-A-.

[5] Ilya Loshchilov and Frank Hutter. *SGDR: Stochastic Gradient Descent with Warm Restarts*. 2017. arXiv: 1608.03983 [cs.LG].

[6] Nikhil Mishra et al. *A Simple Neural Attentive Meta-Learner*. 2018. arXiv: 1707.03141 [cs.AI].

[7] *openreview*. URL: https://openreview.net/group?id=ICLR.cc/2018/Conference.

[8] *openreview*. URL: https://openreview.net/group?id=ICLR.cc/2020/Conference#accept-poster.

[9] *openreview*. URL: https://openreview.net/.

[10] *openreview-py*. URL: https://openreview-py.readthedocs.io/en/latest/.

[11] Pablo Ruiz. *Understanding and visualizing ResNets*. 2018. URL: https://towardsdatascience.com/understanding-and-visualizing-resnets-442284831be8.

[12] *scraper*. URL: https://github.com/Marcus9512/Judging-an-academic-paper-by-its-cover/blob/master/src/Tools/open_review_dataset.py.

[13] Connor Shorten and Taghi M Khoshgoftaar. "A survey on Image Data Augmentation for Deep Learning". eng. In: *Journal of big data* 6.1 (2019), pp. 1–48. ISSN: 2196-1115.

[14] M. Skorikov and S. Momen. "Machine learning approach to predicting the acceptance of academic papers". In: *2020 IEEE International Conference on Industry 4.0, Artificial Intelligence, and Communications Technology (IAICT)*. 2020, pp. 113–117. DOI: 10.1109/IAICT50021.2020.9172011.

[15] AI Technology Industry Review — syncedreview.com. *Medium2018*. URL: https://medium.com/syncedreview/iclr-2018-kicks-off-in-vancouver-f3a99bab70e0.

[16] AI Technology Industry Review — syncedreview.com. *Syncedreview2020*. URL: https://syncedreview.com/2019/12/20/iclr-2020-accepted-papers-announced/.

[17] Sasha Targ, Diogo Almeida, and Kevin Lyman. *Resnet in Resnet: Generalizing Residual Architectures*. 2016. arXiv: 1603.08029 [cs.LG].

[18] UN. *Promote sustained, inclusive and sustainable economic growth, full and productive employment and decent work for all*. URL: https://sdgs.un.org/goals/goal8.

[19] UN. *Reduce inequality within and among countries*. URL: https://sdgs.un.org/goals/goal10.

# 6 Appendix

In this section the loss curves from Resnet32 trained on ICLR  MIDL are presented. Note that figure 22 represents the evaluation curve with no visual bias, which is more unstable then the others.

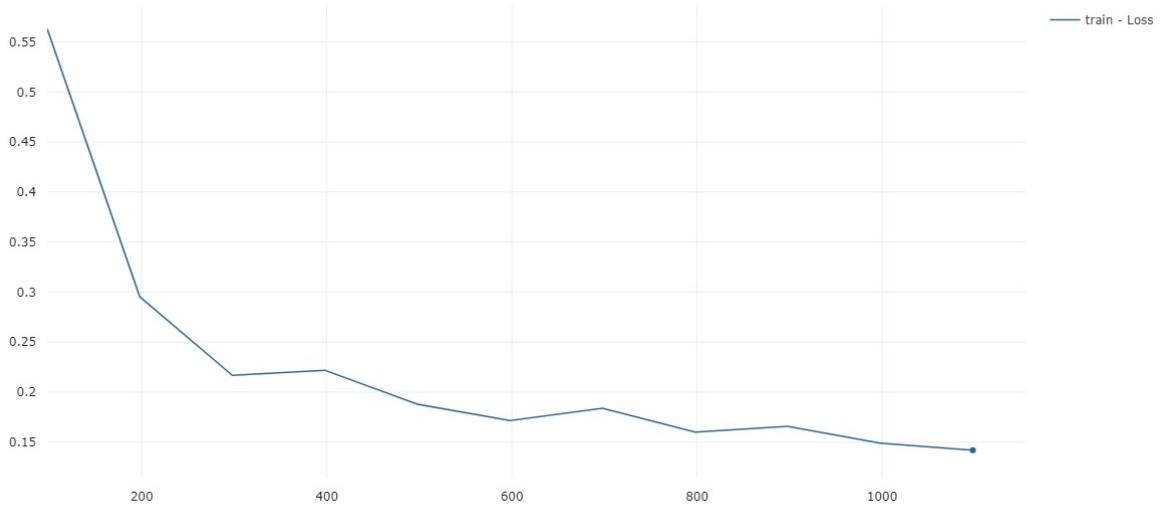



**Figure 15:** The train loss for Resnet32 trained with data included frontpage and header.

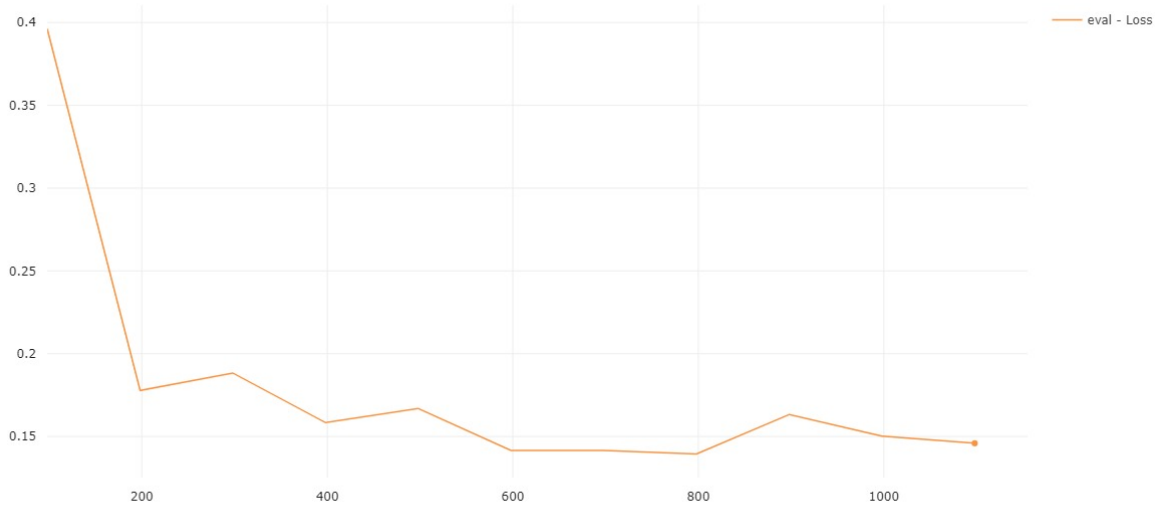



**Figure 16:** The eval loss for Resnet32 trained with data included frontpage and header.

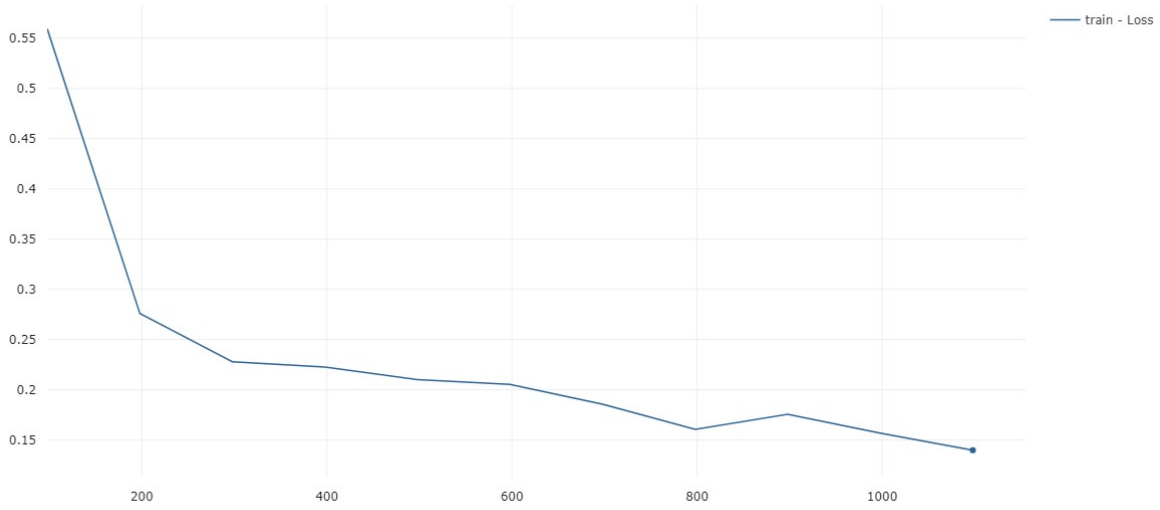



**Figure 17:** The train loss for Resnet32 trained with data front page excluded.

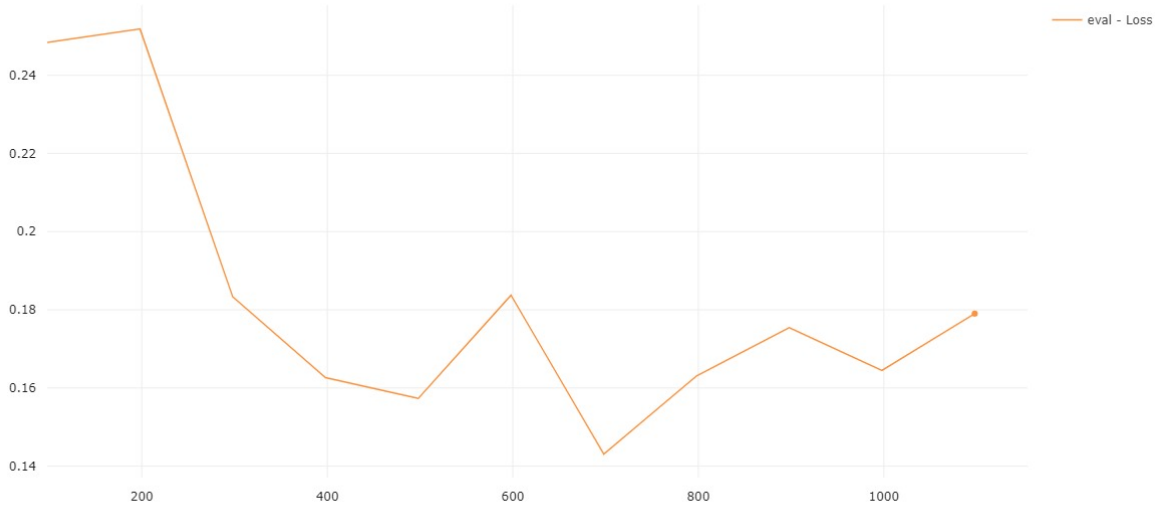



**Figure 18:** The eval loss for Resnet32 trained with data front page excluded.

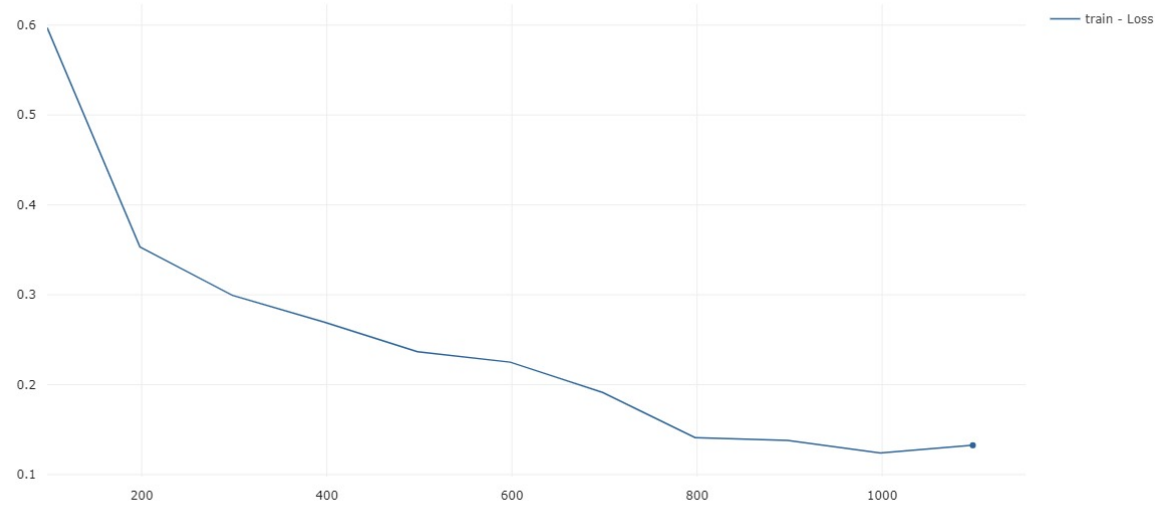



**Figure 19:** The train loss for Resnet32 trained with header excluded.

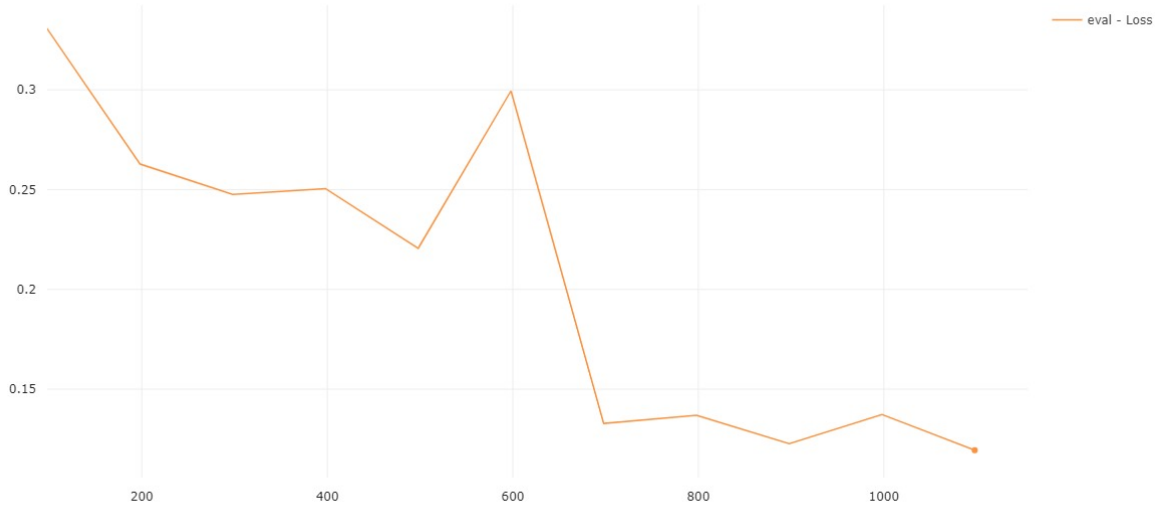



**Figure 20:** The eval loss for Resnet32 trained with header excluded.

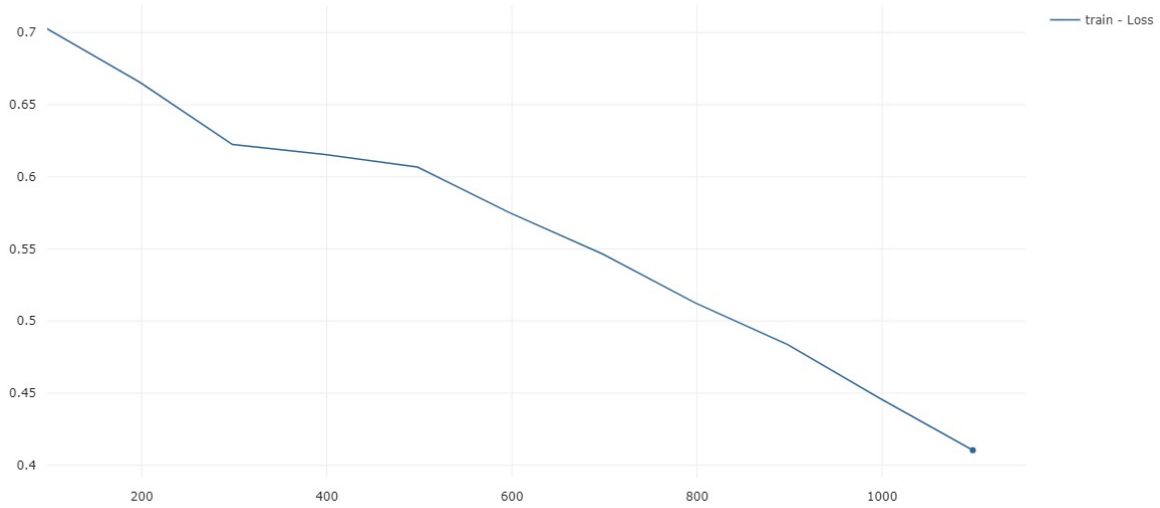



**Figure 21:** The train loss for Resnet32 trained with header and front page excluded.

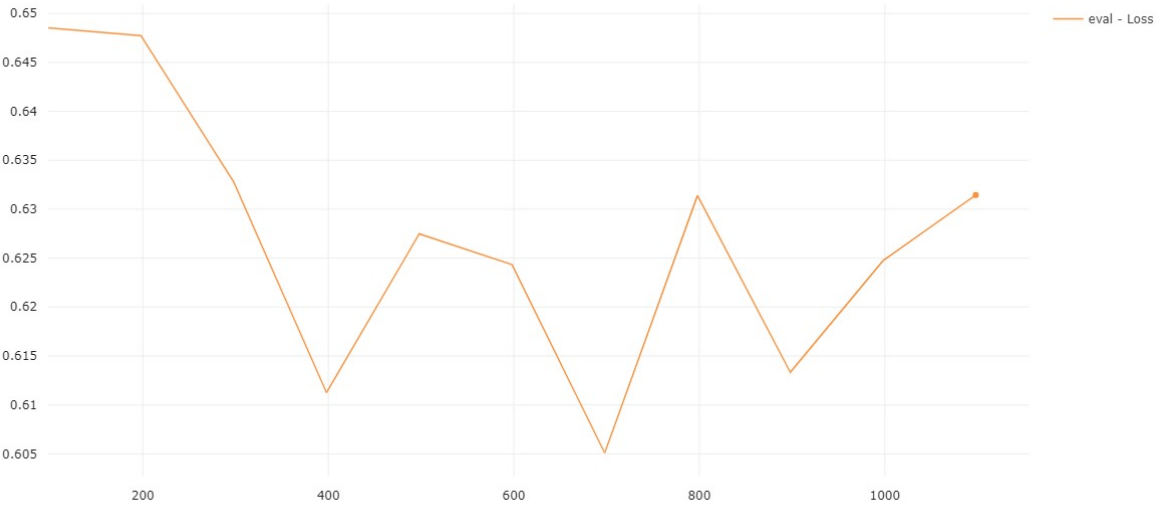



**Figure 22:** The eval loss for Resnet32 trained with header excluded and front page excluded.