Experimenting with Part Templates in Interpretable Convolutional Neural Networks

Jeremy Ong
tto@andrew.cmu.edu

William Liu
wxl@andrew.cmu.edu

1 Introduction

Recent advancements in CNNs have created powerful, deep architectures that can perform accurately given large amounts of training data. However due to the depth and complexity of modern CNN architectures and the difficulty of interpreting them, they are often treated as black box functions. As such they are difficult to tune and reason over. This can be alarming when deep CNNs are used for sensitive applications like self driving cars or determining criminal sentences.

Our project focuses on modifications of convolutional neural networks to be more interpretable. In an interpretable convolutional neural network, each filter in the higher convolutional layers targets a specific feature part of the input object. We will be primarily building off of the work of Q. Zhang *et al.*[11] by experimenting with the various designs of the "part templates". This will be discussed in more detail in section 4.

We perform modifications on four typical CNNs: AlexNet [5], VGG-M [8], VGG-S [8], and VGG-16 [8] and measure/observe the effect of our modifications on the interpretability, accuracy, and convergence behavior of the models. We experiment with single-category classification using custom "part templates." We measure/observe interpretability by calculating the locational instability metric which is the variation in the distance between inferred parts and object landmarks and by visualizing receptive fields of interpretable filters. This is explained in more detail in section 4.

Our project requires the use of data with ground-truth labeled parts, so we utilize the Pascal VOC Part dataset [3] which contains 107 object landmarks in six animal categories: bird, cat, cow, dog, horse, sheep.

As CNN interpretability is still a relatively new direction of research in the space of CNNs, we wish to provide an adequate exploratory characterization of what is possible to build upon the current interpretability characterization methods. The primary contribution of this work will be showing the results of those explorations to give insight into how flexible the variations of interpretable part templates are and their utility for interpretable CNNs.

2 Background

The results from the midway report are based on single class classification on each of the six different animals in the Pascao VOC Part dataset [4]. We include results from the four previously listed CNNs and their modified versions. The modified versions are exactly the same used in the work of Q. Zhange *et al.*[11] which use the following L1-norm part template:

$$T_\mu = (t_{ij}^+), t_{ij}^+ = \tau \cdot \max(1 - \beta \frac{\|[i,j] - \mu\|_1}{n}, -1)$$

.

This part template is visualized in figure 2

Later, we will experiment with custom part templates and compare with the midway report results.

10707 Course Project Final Report (2019).

We have recorded the location instability measures in table 1 and classification accuracies in table 2. In table 1, L1 means that those models have been modified with the L1-norm templates which is the term we use to refer to the templates used by Zhang *et al.*in [11] to modify the original networks. It can be seen that the interpretable networks have much better location instability scores (lower is better) but have a lower performance in classification accuracy compared to the original networks.

Model	bird	cat	cow	dog	horse	sheep	Avg.
AlexNet	0.146	0.133	0.145	0.128	0.143	0.138	0.139
AlexNet (L1)	0.095	0.094	0.099	0.092	0.098	0.098	0.096
VGG-16	0.138	0.136	0.145	0.138	0.141	0.137	0.139
VGG-16 (L1)	0.099	0.096	0.113	0.092	0.098	0.096	0.099
VGG-M	0.145	0.138	0.144	0.131	0.146	0.142	0.141
VGG-M (L1)	0.099	0.095	0.097	0.090	0.093	0.097	0.095
VGG-S	0.138	0.132	0.139	0.128	0.151	0.142	0.138
VGG-S (L1)	0.094	0.092	0.094	0.093	0.098	0.099	0.095

Table 1: Midway Report Results: Location instability

AlexNet	93.2	VGG-16	95.8
AlexNet (L1)	91.6	VGG-16 (L1)	93.1
VGG-M	96.7	VGG-S	95.5
VGG-M (L1)	94.2	VGG-S (L1)	92.3

Table 2: Midway Report Results: Classification accuracy

3 Related Work

3.1 Understanding Large CNNs

There has been some recent work in understanding large CNNs.

An area of research is the visualization of filters. [9] introduces a visualization technique that can be used to give insight into how the CNN works. There is also some work in diagnosing CNNs to make them less of a black-box. [6] finds regions of interest in DNNs and associates dominant classes with these regions of interest. At another level, some work is being done to learn better representations of CNNs that will make it easier to understand them. In [7] they selectively penalize gradients in order to constrain explanations for how the CNN arrives at its prediction, but the training requires the annotation of parts.

The CAR index approach [1] uses network compression through pruning to prune away the convolution filters that contribute least to the final classification output. Their results show that their compression is able to prune away redundant convolution filters. Though it is difficult to say whether or not this is quantifiably an increase in the interpretability of these filters or the networks overall.

3.2 Interpretability Metrics

There have been multiple approaches to quantifying the interpretability of CNNs as well. Generally the interpretability measures are not computed over an entire network, but rather over individual convolution filters.

Alignment with human-interpretable concepts [2] is not an immediately objectively quantifiable metric. As such the metrics proposed in the paper from David Bau and Bolei Zhou *et al.*had to be generated using Amazon Mechanical Turk (AMT). However with human variability, it is difficult to exactly measure alignment with human-interpretable concepts.

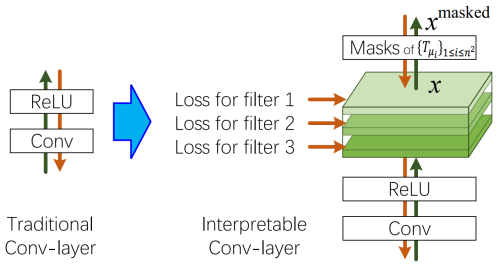

Figure 1: An Interpretable convolutional layer. Figure borrowed from [11]

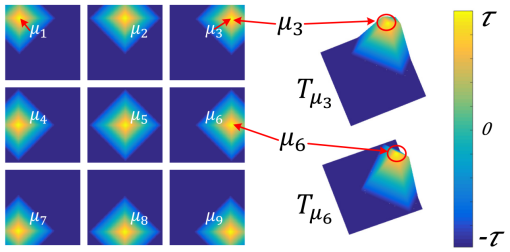

Figure 2: The templates T_{u_i} using the L1 norm distance. Figure borrowed from [11]

4 Methods/Models

This project builds off of the work of Q. Zhang *et al.*[11] which alters the training of the CNN to arrive at a more interpretable CNN without requiring the annotation of parts for training. We will experiment with custom "part templates." Which will be explained in this section.

We build off of Zhang's code available on GitHub at `https://github.com/zqs1022/interpretableCNN` which alters the following four CNN architectures to be more interpretable: AlexNet [5], VGG-M [8], VGG-S [8], and VGG-16 [8].

4.1 CNN Architecture Modification

The modification of the CNN architectures is done by first converting the highest convolutional layer into an interpretable convolutional layer as shown in figure 1. Then, another interpretable convolutional layer is appended to this converted layer such that the model has two interpretable convolutional layers. The hyperparameters for the filters are defined such that if the original network has k channels of $n \times n$ images coming out of the highest convolutional layer, the modified CNN has k channels of $n \times n$ images coming out of each of the new interpretable convolutional layers. Interpretable convolutional layers make use of the templates shown in figure 2 for the Masks layer in figure 1 in order to focus on specific object parts. There is a template centered around each coordinate of the outputted $n \times n$ channels i.e. we have $T_{u_i} \forall i \in [1, n \times n]$. For each image a different template is chosen for each outputted feature map x of the "Relu" part based on the most activated coordinate. The details of the forward propogation and backward propogation are explained in [11].

4.2 Custom Part Templates

We are interested in how changing the structure of the part templates affects the interpretability of the model as well as the performance of the model. Our hope is that we can better understand how Zhang's interpretable convolutional networks work by experimenting with custom part templates.

The part template shapes that we experiment with are as follows:

- $T_\mu = (t_{ij}^+), t_{ij}^+ = \tau \cdot \begin{cases} 1 & \beta \frac{\|[i,j]-\mu\|_1}{n} < \gamma \\ -1 & \text{otherwise} \end{cases}$
- $T_\mu = (t_{ij}^+), t_{ij}^+ = \tau \cdot \max(1 - \beta \frac{\|[i,j]-\mu\|_2}{n}, -1)$
- $T_\mu = (t_{ij}^+), t_{ij}^+ = \tau \cdot \begin{cases} 1 & \beta \frac{\|[i,j]-\mu\|_2}{n} < \gamma \\ -1 & \text{otherwise} \end{cases}$

We will refer to these part templates as the uniform L1-norm template, L2-norm template, and uniform L2-norm template respectively. The original template used in [11] will be referred to as the L1-norm template. We also refer to the non-uniform templates as the gradient templates.

Essentially for the uniform templates we are using the same L1-norm and L2-norm distances but all values throughout the template are either a value of -1 or a uniform value of 1.

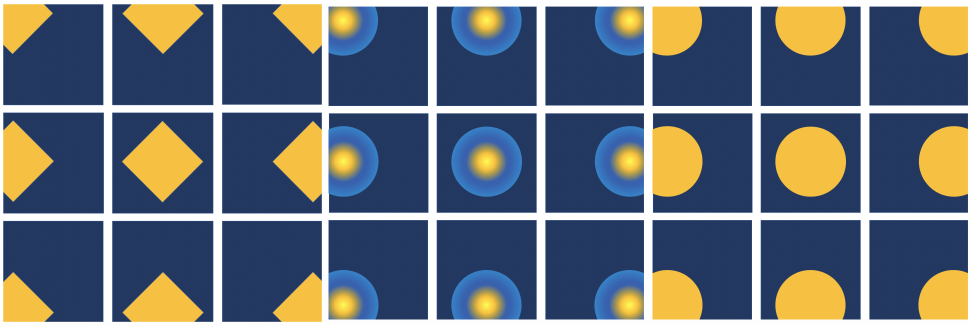

Figure 3: Uniform L1-Norm Figure 4: L2-Norm Figure 5: Uniform L2

Visualization of our modified part templates

In our experiments we use $\beta = 2$ and $\gamma = 2$.

The reasoning behind using uniform part templates is that the uniform part templates will cause the resulting masked filter outputs to have higher activation which will more strongly propogate into the loss of the filter due to the Mask layer. The hope is that this will cause the filters to have a more intense update due to the Mask layer and that this will cause the network to better focus on various animal parts.

The reasoning behind experimenting with the L2-norm template is that the L2-norm template has a more gradual decrease in intensity as we move further away from the reference point compared to the L2-norm template. When focusing on animal parts, it seems more natural to focus on a part using a template using the L2-norm like that in figure 4 rather than a template that uses the L1-norm like that in figure 2 since the L1-norm makes the template diamond shaped.

A visual representation of what our modified part templates look like can be found in figures 3, 4, and 5.

4.3 Metrics

Our performance measure for the interpretability of the networks is location instability[10]. This will only be measured for the higher layers of the CNN as they are more likely to represent object parts. Location instability is the deviation of the distance between the inferred parts of the higher layer filters and object landmarks. For a given image I and filter f, the normalized distance between the inferred part and an object landmark can be expressed as $d_I(p_k, f) = \frac{\|\mathbf{p}_k - \mathbf{p}(f)\|}{\sqrt{w^2+h^2}}$ Where $\mathbf{p}_k$ is the kth object landmark and $\mathbf{p}(f)$ is the center of of the filter f's receptive field field on the original image and w and k are the original image's width and height. The receptive field is determined using the method in [2]. The *relative location deviation* is then calculated as $D_{f,k} = \sqrt{\text{var}_I\left[d_I(p_k, f)\right]}$ and the location instability is calculated as $\text{mean}_f \text{mean}_{k=1}^{K} D_{f,k}$ where K is the number of landmarks.

The chosen object landmarks to compute the location instability are those with the top-100 activation values.

We also observe the effect of our various custom part templates on the classification accuracy. And the effect of the custom part templates on the convergence rate of the training.

Additionally we visualize the receptive fields using the method used in [12] to get a better idea of the behavior of the interpretable convolutional neural networks after our modifications.

5 Results

5.1 Location Instability

We report the location instabilities of each of the networks on each of the animal categories in table 3.

Model	bird	cat	cow	dog	horse	sheep	Avg.
AlexNet (Uni-L1)	0.142	0.138	0.145	0.127	0.126	0.125	0.134
AlexNet (L2)	0.094	0.094	0.096	0.093	0.097	0.095	0.095
AlexNet (Uni-L2)	0.147	0.124	0.135	0.119	0.148	0.131	0.134
VGG-16 (Uni-L1)	0.135	0.116	0.128	0.127	0.141	0.145	0.132
VGG-16 (L2)	0.096	0.090	0.098	0.097	0.094	0.098	0.096
VGG-16 (Uni-L2)	0.143	0.128	0.123	0.117	0.135	0.132	0.130
VGG-M (Uni-L1)	0.136	0.122	0.135	0.118	0.132	0.138	0.130
VGG-M (L2)	0.100	0.092	0.094	0.095	0.098	0.093	0.095
VGG-M (Uni-L2)	0.144	0.115	0.139	0.121	0.136	0.128	0.131
VGG-S (Uni-L1)	0.126	0.135	0.149	0.111	0.142	0.139	0.134
VGG-S (L2)	0.098	0.091	0.095	0.091	0.093	0.094	0.094
VGG-S (Uni-L2)	0.137	0.118	0.135	0.119	0.148	0.133	0.132

Table 3: Final Report Results: Location instability

Figure 6: Generated Receptive Fields Resulting from Using Different Part Templates with AlexNet (In each row, for the same animal, the RF from the same filter is displayed)

We hypothesized that the uniform part templates would benefit the interpretability of the models resulting in lower location instability scores, but this disagrees with the results because the models using uniform templates had higher location instability measures compared to the models using gradient templates.

We hypothesized that the the L2-norm template models would have better interpretability than L1-norm template models because they more naturally align with animal parts. The models using L1-norm templates seem to have all performed better than the models using L1-norm templates as shown in table 4. But the improvement is very minute and may just be due to luck.

Model	L1-norm	L2-norm	improvement
AlexNet	0.096	0.095	0.001
VGG-16	0.099	0.096	0.003
VGG-M	0.095	0.095	0.000
VGG-S	0.095	0.094	0.001

Table 4: Average locational instabilities, comparing L1-norm and L2-norm templates

We also note that between the different animals, there seems to be a trend where single-class classification on cats and dogs has better locational instability as can be seen in table 3.

Additionally as shown in figure 6 the receptive fields of the CNN filters show similar characteristics across the rows for the L1-norm and L2-norm template models which agrees with their location instability measures. On the other hand, for the uniform templates, the receptive fields appear to be less precise and do not overlay consistent animal parts.

5.2 Classification Accuracy

The classification accuracies are summarized in table 5.

AlexNet (Uni-L1)	92.4	VGG-16 (Uni-L1)	93.1
AlexNet (L2)	91.5	VGG-16 (L2)	92.1
AlexNet (Uni-L2)	92.5	VGG-16 (Uni-L2)	93.6
VGG-M (Uni-L1)	94.9	VGG-S (Uni-L1)	93.4
VGG-M (L2)	93.8	VGG-S (L2)	93.1
VGG-M (Uni-L2)	95.5	VGG-S (Uni-L2)	94.9

Table 5: Final Report Results: Classification accuracy

The classification accuracies of the uniform L1 and L2 part templates are consistently higher than the gradient part templates. While the difference is not significant, every single uniform part template on every network architecture performs better than its gradient counterpart.

5.3 Convergence Rate

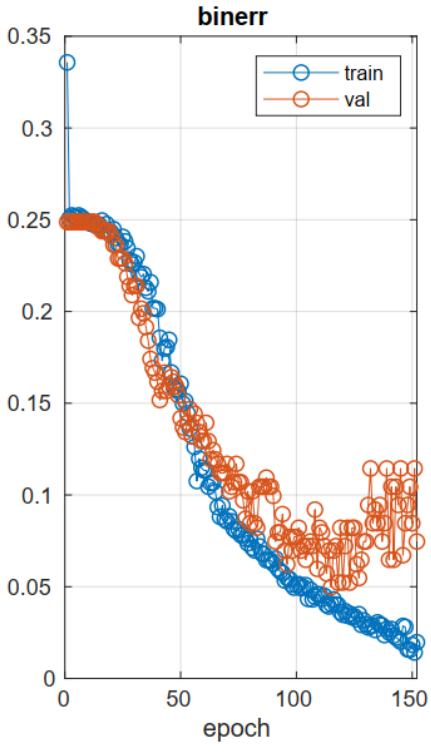

Figure 7: Binary Error for Non-uniform Template on AlexNet

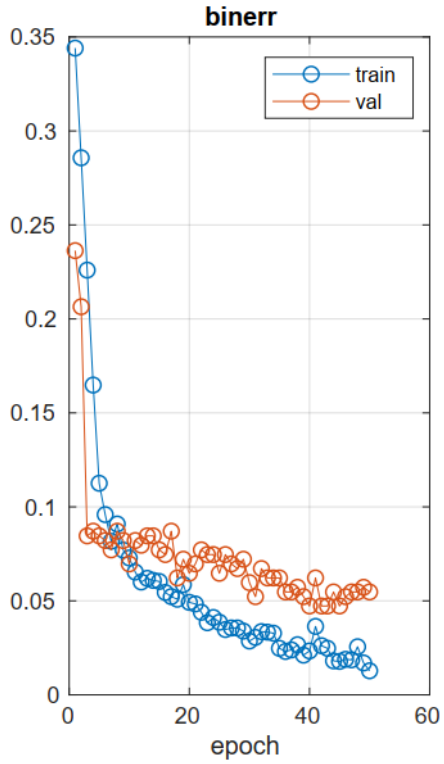

Figure 8: Binary Error for Uniform Template on AlexNet

We can see that the uniform L1 and L2 norm distance part templates converged very quickly at around 50 epochs, whereas the gradient part templates converged at a much slower rate at around 115 epochs. These rates of convergence can be observed in figures 7 and 8.

6 Discussion and Analysis

6.1 Location Instability

In section 5 it was seen that the models using uniform templates had worse location instability measures than those using gradient templates. We can look at how the backprop update of interpretable filters is calculated to explain this.

From Q. Zhang *et al.*[11], we know that the gradient with respect to a feature map x (this is the same x in figure 1) for the interpretable filters is expressed as the sum of the partial derivative for the local filter loss and the partial derivative for the classification loss:

$$\frac{\partial \textbf{Loss}}{\partial x_{ij}} = \lambda \frac{\partial \textbf{Loss}_f}{\partial x_{ij}} + \frac{1}{N} \sum_{i=k}^{N} \frac{\partial \mathbf{L}(\hat{y}_k, y_k^*)}{\partial x_{ij}} \tag{1}$$

where the local filter loss for the interpretable filter is expressed as

$$\textbf{Loss}_f = -MI(\mathbf{X}; \mathbf{T}) \tag{2}$$

$\frac{\partial \textbf{Loss}}{\partial x_{ij}}$ is eventually propagated to the interpretable filter.

In equation 1, $\mathbf{X}$ is the set of feature maps of a filter after a ReLu operation and $\mathbf{T}$ is the set of $n^2 + 1$ template candidates which includes a negative template $\mathbf{T}^-$ which is only comprised of the value -1 and is mapped to images that are not part of the target class. $\mathbf{T}^-$ is not used in the forward propagation. This is explained in further detail in [11].

The intuition behind $\textbf{Loss}_f$ is that for every image in the target class, we want the image to definitively map to a certain template in $\mathbf{T}$. So we want a high mutual information between $\mathbf{X}$ and $\mathbf{T}$.

So for uniform templates, since they uniformly activate such a large part of the feature map, the value of $\frac{\partial \textbf{Loss}_f}{\partial x_{ij}}$ is small compared to when a gradient template is used. This is because moving around the uniform template has less of an impact than moving around a gradient template. This causes the update value calculated with equation 1 to be smaller for the uniform template than for the gradient template. This causes the contribution of the local filter loss to possibly be overshadowed by the contribution of the classification loss.

It was also seen that the models using the L2-norm template performed slightly better than those using the L2-norm template. The improvement is very slight and could be due to luck. The magnitude of improvement is not enough to conclude anything. But perhaps this is because it is more natural to use an L2-norm template with the animal parts.

6.2 Classification Accuracy

As observed in section 5, the uniform templates seems to have better classification accuracy than the gradient templates. This can be explained by the reasoning in section 6.1. The contribution of the local filter loss to the backprop update is so small that the contribution of the classification loss dominates the overall backprop update causing the final accuracy to be even better.

6.3 Convergence Rate

It was also seen in section 5 that the model with the uniform template converged much faster than the model with the gradient template. This is probably also due to the smaller contribution of the local filter loss as explained in section 6.1. The gradients of the local filter loss and the classification loss do not seem to go in the same direction.

6.4 Tradeoff Between Interpretability and Classification

The results suggest that as the model tries to optimize towards being more interpretable, it competes with the optimization that attempts to improve the classification accuracy. The results suggest this because across all the experiments, there is a trend where when the location instability is lower, the classification accuracy is lower.

6.5 Limitations

One limitation of the approach of part interpretability is that part interpretability is dependent on the existence of semantic parts in the image that are about the same size of the templates used. For example, if we were working with a CNN that had to classify solid color images, including layers to focus the filters towards semantic parts in the image would be useless.

There is also a limitation with the location instability metric. We use the intuition that a lower location instability score implies higher interpretability, but this assumes that semantically segmented parts, once they are defined, don't differ much in distance from landmarks. But a weakness in this metric can be seen when considering using the head and tail of a snake as landmarks. Across different images it is easy to imagine how these animal parts can be varying distances from other parts of the snake. In the dataset that we used, this assumption can be safely made.

7 Future Work

Extending past this work, there are a few possible directions that we believe would be interesting to explore.

The first is the obvious testing of different part template shapes. Our results have already shown that there is consistent variability between the uniform part templates versus the gradient part templates. There is merit in further testing of part template shapes as further exploration of the possible variations of using interpretable filters beyond what this work presents. It would be interesting to see if custom part templates could be tailored to the datasets. It would also be worthwhile conducting experiments which use other kinds of interpretability measures that don't use the same assumptions as location instability.

The second is using the part interpretability metrics, along with the classification error for the tuning of hyperparameters. This is an interesting space as it provides even more utility for improving the interpretability of CNNs.

References

[1] Reza Abbasi-Asl and Bin Yu. Interpreting convolutional neural networks through compression. *CoRR*, abs/1711.02329, 2017.

[2] David Bau, Bolei Zhou, Aditya Khosla, Aude Oliva, and Antonio Torralba. Network dissection: Quantifying interpretability of deep visual representations. In *Computer Vision and Pattern Recognition*, 2017.

[3] Xianjie Chen, Roozbeh Mottaghi, Xiaobai Liu, Sanja Fidler, Raquel Urtasun, and Alan L. Yuille. Detect what you can: Detecting and representing objects using holistic models and body parts. *CoRR*, abs/1406.2031, 2014.

[4] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman. The PASCAL Visual Object Classes Challenge 2007 (VOC2007) Results. http://www.pascal-network.org/challenges/VOC/voc2007/workshop/index.html.

[5] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton. Imagenet classification with deep convolutional neural networks. In *Proceedings of the 25th International Conference on Neural Information Processing Systems - Volume 1*, NIPS'12, pages 1097–1105, USA, 2012. Curran Associates Inc.

[6] Devinder Kumar, Alexander Wong, and Graham W. Taylor. Explaining the unexplained: A class-enhanced attentive response (CLEAR) approach to understanding deep neural networks. *CoRR*, abs/1704.04133, 2017.

[7] Andrew Slavin Ross, Michael C. Hughes, and Finale Doshi-Velez. Right for the right reasons: Training differentiable models by constraining their explanations. *CoRR*, abs/1703.03717, 2017.

[8] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *CoRR*, abs/1409.1556, 2014.

[9] Matthew D. Zeiler and Rob Fergus. Visualizing and understanding convolutional networks. *CoRR*, abs/1311.2901, 2013.

[10] Quanshi Zhang, Ruiming Cao, Feng Shi, Ying Nian Wu, and Song-Chun Zhu. Interpreting cnn knowledge via an explanatory graph. In Sheila A. McIlraith and Kilian Q. Weinberger, editors, *AAAI*, pages 4454–4463. AAAI Press, 2018.

[11] Quanshi Zhang, Ying Nian Wu, and Song-Chun Zhu. Interpretable convolutional neural networks. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018.

[12] Bolei Zhou, Aditya Khosla, Agata Lapedriza, Aude Oliva, and Antonio Torralba. Object detectors emerge in deep scene cnns. *arXiv preprint arXiv:1412.6856*, 2014.