# Receptive Field and Neural Networks Representations

**Luis Alfredo Avendaño Muñoz.** *
School of Computing
University of Leeds
Leeds, United Kingdom
sclaam@leeds.ac.uk

## Abstract

## 1 Introduction

Neural networks have emerged as the standard approach for a wide range of applications, ranging from image recognition [Deng et al., 2009] to natural language processing [Devlin et al., 2019] and speech synthesis [LeCun et al., 2015; van den Oord et al., 2016]. Neural networks owe their expressiveness to the ability to learn hierarchical features and representations form the data that are particularly suited for the task at hand [Goodfellow et al., 2016]. These features are influenced by the highly non-linear.

The reason that make neural networks so good is the features that they learn from the data ([Zhou et al., 2015]), They learning general features in the first layers while in deeper layer they integrate and combine the simpler features and create more complex and detailed features. One aspect that greatly affects these representations, as well as training dynamics is the receptive field of neural networks, yet, its effect on the representations remains not fully understood.

In this paper we systematically explore the relationship that the receptive field has with the representations of neural networks along with the loss landscape these models. We show that if we manipulate the receptive field by changing the kernel size of the first maxpooling layer leaving every other component static we can demonstrate that the models losses generalization capacity as we increase the receptive field. Surprisingly, for high levels of pruning, it turns out that the negative effect on accuracy have an inverse relationship with the receptive field. We show that the receptive fields greatly affects the loss landscape of the models as shown by the Hessian spectra at initialisation.

We empirically validated our results with two different type of models with similar number of parameters (VGG-like and Resnet50) and two different datasets with different image sizes,( CIFAR10 and Tiny imagenet)

Our contributions are:

- We show that the receptive field greatly affects the loss landscape defining the type of features that of these models an be learning and in consequence affecting the robustness of the networks against pruning
- We link the loss landscape to the features learned by the network and their robustness to pruning.

*Use footnote for providing further information about author (webpage, alternative address)—*not* for acknowledging funding agencies.

Preprint. Under review.

## 2 Related Work

**Receptive field literature**

- Luo et al. [2016]
- object detectors Zhou et al. [2015]
- Interpretation of ResNet by Visualization of Preferred Stimulus in Receptive Fields Kobayashi and Shouno [2020]

**Landscape investigation**

- On the Relation Between the Sharpest Directions of DNN Loss and the SGD Step Length Jastrzębski et al. [2019]

## 3 Experimental results

In this section we empirically investigate the effect that the receptive field has in a network's trainability and prunability. We Statistically validate our results on VGG and ReseNet50 architecture and in CIFAR10 and Tiny ImageNet datasets.

Here I never talked of how I calculated the receptive field, which I used a libary that calculates the gradient projection in a dummy input space with a projected gradient of 1 in the middle of all the fueature maps of the last convolutional layer for the two architectures.

### 3.1 Experimental settings

We used a custom implementation of the ResNet50 and VGG models with a modified maxpooling layer after the first convolutional layer for manipulating the receptive field. Implementation details are in **??**. The dataset are CIFAR10 and Tiny ImageNet. The training hyperparameters are the following:

- Epochs: 200
- Optimizer: SGD
- Learning rate: 0.0001
- Learning rate schedule: Cosine annealing with $T_{max} = 200$
- Momentum: 0.9
- Weight Decay: $5 \times 10^{-5}$
- Gradient clipping: 0.1

### 3.2 Manipulating the Receptive Field

There is plentiful of ways for Manipulating the receptive field. It is known that the presence of skip connections affect the receptive fields along with depth and dilation on convolutions. We wanted to alter the receptive field minimum alterations to the rest of the networks leaving all layers with the exact same number of parameters. We placed a maxpooling layer just after the first convolutional layer on both architectures and we changed the kernel size of that layer. That is the only difference between the different models on each experiment. For each architecture-dataset-receptive field combination we trained 5 models and the result are shown in 3.3.

### 3.3 Receptive Field, Accuracy and Loss Landscape

Here we show the dense accuracy after training the models and its one-shot pruned accuracy. One interesting observation is that larger receptive field correlates with lower accuracy but simultaneously its one-shot pruning performances is better than those for smaller receptive fields. This behaviour generalises across all the combinations of architecture-dataset. In Section 4.1 we show that the current trend in accuracy (for dense and prune version) only arises for sufficiently large pruning rate (>0.8). We also fine-tuned the models to test their real-world application capabilities as seen in Section 4.1.1 but here we only show the pruning rate of 0.9 as means to show the one-shot behaviour of these

networks in order to understand their one-shot potential. For each one of the combinations shown here we trained and pruned 5 models, the values presented are the mean with its standard deviation.

The image size for CIFAR10 and Tiny ImageNet is 32x32 and 64x64 respectively. In both cases the receptive fields of each architecture is greater than the size of the image.

| **VGG** | | | |
|---|---|---|---|
| **Receptive Field** | **Dense Test Accuracy** | **Pruned Test Accuracy** | **Difference in Accuracy** |
| 181 | $93.5 \pm 0.11$ | $10.9 \pm 2.03$ | $82.6 \pm 2.02$ |
| 359 | $91.1 \pm 0.23$ | $32.4 \pm 15.7$ | $58.7 \pm 15.8$ |
| 537 | $87.8 \pm 0.19$ | $87.6 \pm 0.30$ | $0.18 \pm 0.16$ |
| 715 | $85.8 \pm 0.21$ | $85.8 \pm 0.22$ | $0.07 \pm 0.04$ |
| **Resnet 50** | | | |
| **Receptive Field** | **Dense Test Accuracy** | **Pruned Test Accuracy** | **Difference in Accuracy** |
| 110 | $94.8 \pm 0.29$ | $56 \pm 20.7$ | $38.7 \pm 20.5$ |
| 213 | $94.0 \pm 0.16$ | $91.7 \pm 0.98$ | $2.25 \pm 0.90$ |
| 318 | $92.2 \pm 0.18$ | $91.4 \pm 0.45$ | $0.85 \pm 0.49$ |
| 423 | $90.4 \pm 0.30$ | $90.2 \pm 0.37$ | $0.18 \pm 0.09$ |

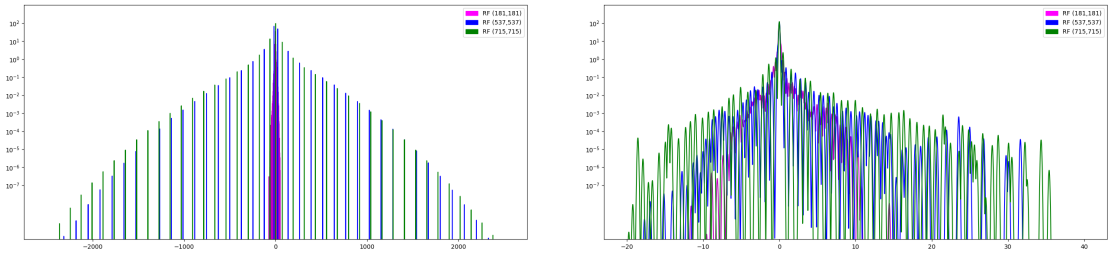
Table 1: **CIFAR10 results:** Here are summarised the results for the experiments performed in CIFAR10. It can be seen that the discrepancy in accuracy between different receptive fields is consistent for these two architectures. Also, as we increase the receptive field we can see that the gap in performance between dense and pruned models diminishes. The pruning rate used is 90%

| **VGG** | | | |
|---|---|---|---|
| **Receptive Field** | **Pruned Test Accuracy** | **Dense Test Accuracy** | **Difference in Accuracy** |
| 181 | $0.75 \pm 0.09$ | $61.5 \pm 0.33$ | $60.8 \pm 0.32$ |
| 359 | $0.63 \pm 0.17$ | $53.2 \pm 0.20$ | $52.6 \pm 0.36$ |
| 537 | $16.7 \pm 7.50$ | $41.0 \pm 1.91$ | $24.3 \pm 5.81$ |
| 715 | $21.8 \pm 6.57$ | $38.5 \pm 1.69$ | $16.7 \pm 7.21$ |
| **Resnet 50** | | | |
| **Receptive Field** | **Pruned Test Accuracy** | **Dense Test Accuracy** | **Difference in Accuracy** |
| 213 | $5.91 \pm 0.89$ | $61.8 \pm 0.40$ | $55.9 \pm 0.99$ |
| 318 | $8.56 \pm 2.66$ | $59.1 \pm 0.36$ | $50.5 \pm 2.55$ |
| 423 | $21.4 \pm 2.87$ | $56.5 \pm 0.27$ | $35.0 \pm 2.99$ |

Table 2: **Tiny ImageNet Results:** Here are summarised the results for the experiments performed in Tiny ImageNet. Similarly to CIFAR10, the trend of diminishing dense accuracy and gap between dense and pruned accuracy as we increase the receptive field is consistent in the two architectures. The pruning rate is 90%

Why do models with large receptive field behave in this manner? One hypothesis is that the loss landscape changes in such a way that makes more difficult for SGD to found a better solution. In Figures 1 and 2 we show the 90 largest eigen values for both models on CIFAR10, before and after training. As we can see, at the beginning of training the Hessian spectra of models with alter receptive field are wider and encompass larger eigen values. This means that the landscape of that models has much more steeper directions of descent (and ascent) making the landscape more chaotic and difficult to traverse than for smaller receptive fields (maybe search a citation that corroborates this?).

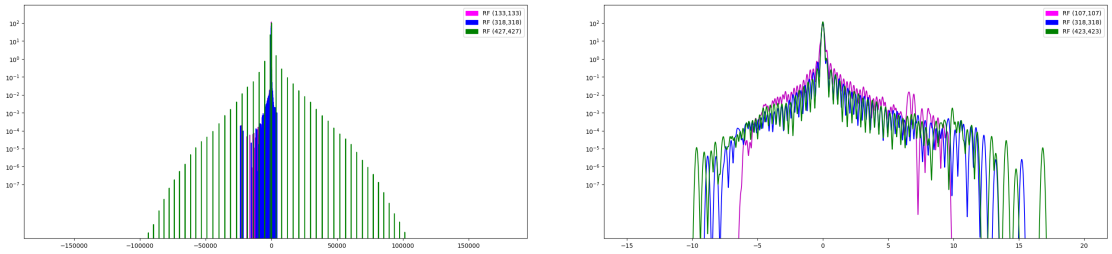
But how does this affect the features or representations that these models learn? Next we show the similarity of the internal representation two different seeds of the ReseNet50 model on CIFAR10., It is observed that the introduction of larger receptive fields results in a divergence of representations within the deepest layers. This phenomenon can be attributed to the heightened "chaotic" nature of the loss landscape, as indicated by the Hessian spectra in Figure 2, particularly in instances where




(a) Before Training (b) After training

Figure 1: Largest 90 eigen values of VGG model on CIFAR10 for different Receptive Fields

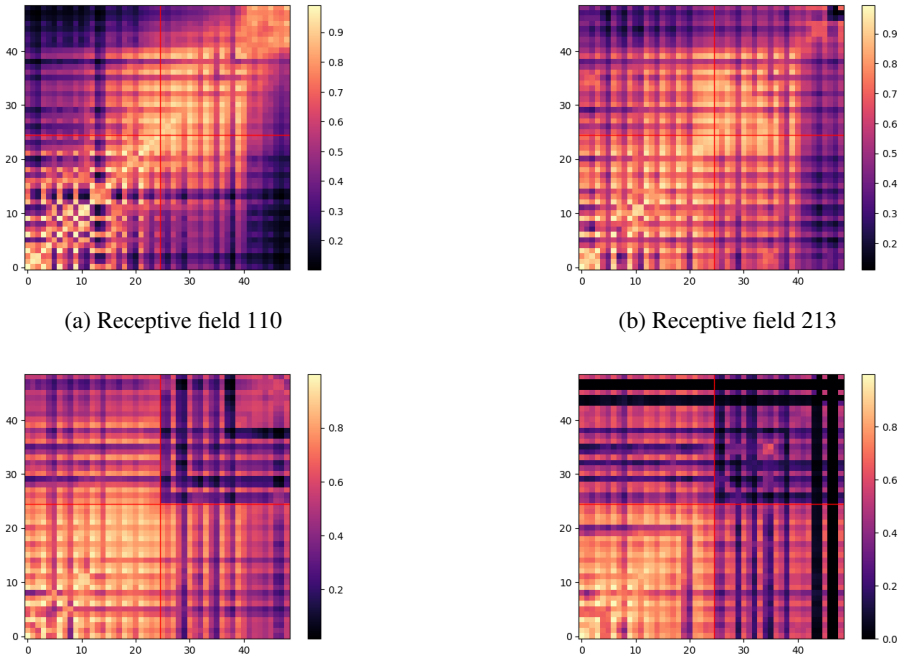
Here I can put lavels like $\lambda$ and $P(\lambda)$. Larger font




(a) Before Training (b) After training

Figure 2: Largest 90 eigen values of ResNet50 model on CIFAR10 for different Receptive Fields

larger receptive fields are employed. Notably, the representations in deeper layers for individual seeds exhibit dissimilarity, as they tend to converge into distinct basins, differing not only from the majority of the network but also from corresponding layers in other seeds. The representation similarly was calculated using the first 1000 images of the test set and we use the Centred linear Kernel Alignment [Kornblith et al., 2019] as similarity measure.



(a) Receptive field 110

(b) Receptive field 213

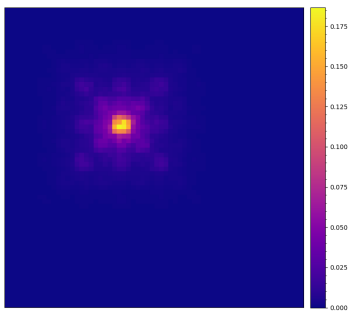
(c) Receptive field 318

(d) Receptive field 423

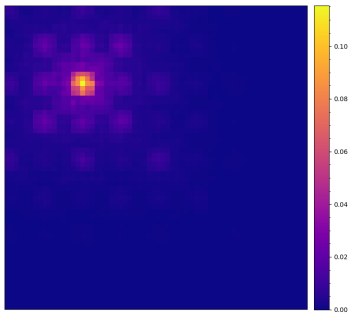
Figure 3: Representation similarly of all layers in ReseNet50 for different receptive fields. The red lines represent the middle layer of the model ($25^{th}$ layer). The largest receptive field (bottom right) has high degree of dissimilarity between its deeper layer representations when compared with the smallest receptive field (top left)

### 3.4 Receptive Field and Pixel Importance

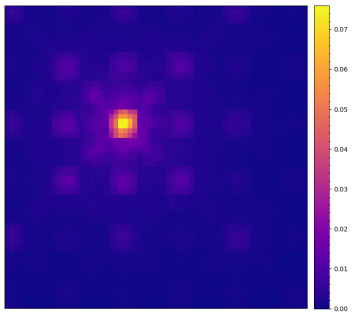
In this section we show the projection of gradient of the last convolutional network into an input space of 64x64. We show that for larger receptive fields the "importance pattern" does not change much but the "importance" decreases from one receptive field to another. This means that even if the model sees similarity for all receptive fields, for larger receptive fields each pixel is less important. We call importance here to the magnitude of the projected gradient on the input space from the last convolutional layer of the randomly initialised model. We place a would be gradient with 1 in the center of all channels of the last convolutional layer and calculate the gradient with respect to a random input with 64x64 dimensions, we take the absolute value of that projected gradient. Then for each receptive field we average 50 of such projections and also take the average of their maximum value. In Figures 4 and 5 we can see the importance pattern for ReseNet50 and VGG. In Figure 5 the check board pattern appears due an odd-sized kernel is applied with an even stride causing what Kim et al. [2023] call *pixel sensitivity imbalance*. In Tables 3 and 4 can be seen the average of the maximum importance value for each receptive field. As we can see the importance for each pixel is decreased as we increase the receptive field. This might explain why having a larger receptive field could hinder performance, since we are paying attention to the same areas of the image but not the same value of attention.



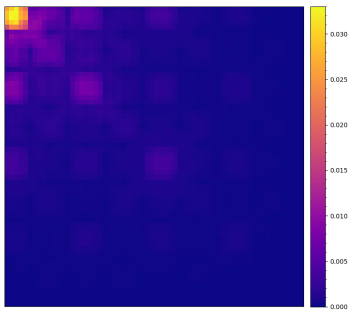

(a) Receptive field 110




(b) Receptive field 213

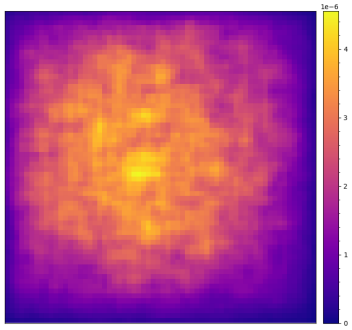



(c) Receptive field 318

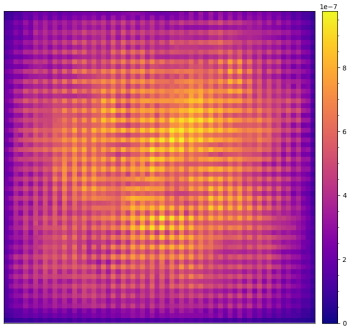



(d) Receptive field 423

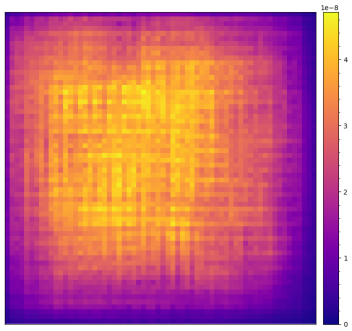
Figure 4: Projection of the gradient into an input space of 64x64 for different levels of receptive field of ReseNet50



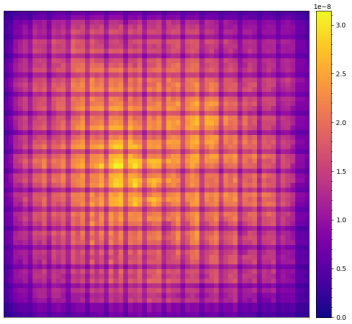
(a) Receptive Field 181



(b) Receptive Field 359



(c) Receptive Field 537



(d) Receptive Field 715

Figure 5: Projection of the gradient into an input space of 64x64 for different levels of receptive field of ReseNet50

| **Receptive Field** | **Maximum Projected Gradient Value** |
|---|---|
| 181 | 1.24e-05 $\pm$ 2.14e-06 |
| 359 | 2.48e-06 $\pm$ 4.88e-07 |
| 537 | 1.32e-07 $\pm$ 2.48e-08 |
| 715 | 7.26e-08 $\pm$ 1.45e-08 |

Table 3: VGG gradient projection on a 64X64 input image

| **Receptive Field** | **Maximum Projected Gradient Value** |
|---|---|
| 110 | 0.322 $\pm$ 0.122 |
| 213 | 0.179 $\pm$ 0.079 |
| 318 | 0.111 $\pm$ 0.039 |
| 423 | 0.064 $\pm$ 0.022 |

Table 4: Maximum projected gradient for ReseNet50

## 4 Conclusions and Future work

In this work we showed how the receptive field affects the loss landscape of neural networks along with its representations and its behaviour on the input image. Is important to note that there are other ways to manipulate the receptive field such as change the kernel size of the convolutional networks,stride and dilation of the convolutional networks. I this work we wanted to have the minimum intervention that would not modify the number of weights of the models as we manipulated the receptive field of the models. Additionally, in our experiments, as we changed the receptive field

also the size of the feature maps were altered. Future work will concentrate on testing other ways to manipulate the receptive field to test the robustness of the findings in this work.

## References


J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. ImageNet: A large-scale hierarchical image database. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pages 248–255, June 2009. doi: 10.1109/CVPR.2009.5206848.

J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics. doi: 10.18653/v1/N19-1423.

I. Goodfellow, Y. Bengio, and A. Courville. *Deep Learning*. MIT Press, 2016.

S. Jastrzębski, Z. Kenton, N. Ballas, A. Fischer, Y. Bengio, and A. Storkey. On the Relation Between the Sharpest Directions of DNN Loss and the SGD Step Length, Dec. 2019.

B. J. Kim, H. Choi, H. Jang, D. G. Lee, W. Jeong, and S. W. Kim. Dead pixel test using effective receptive field. *Pattern Recognition Letters*, 167:149–156, Mar. 2023. ISSN 0167-8655. doi: 10.1016/j.patrec.2023.02.018.

G. Kobayashi and H. Shouno. Interpretation of ResNet by Visualization of Preferred Stimulus in Receptive Fields, July 2020.

S. Kornblith, M. Norouzi, H. Lee, and G. Hinton. Similarity of Neural Network Representations Revisited, July 2019.

Y. LeCun, Y. Bengio, and G. Hinton. Deep learning. *nature*, 521(7553):436–444, 2015.

W. Luo, Y. Li, R. Urtasun, and R. Zemel. Understanding the Effective Receptive Field in Deep Convolutional Neural Networks. In *Advances in Neural Information Processing Systems*, volume 29. Curran Associates, Inc., 2016.

A. van den Oord, S. Dieleman, H. Zen, K. Simonyan, O. Vinyals, A. Graves, N. Kalchbrenner, A. Senior, and K. Kavukcuoglu. WaveNet: A Generative Model for Raw Audio. *arXiv:1609.03499 [cs]*, Sept. 2016.

B. Zhou, A. Khosla, A. Lapedriza, A. Oliva, and A. Torralba. Object Detectors Emerge in Deep Scene CNNs, Apr. 2015.


## Supplementary Material

### 4.1 Model Implementation details

Here are the implementation details.

#### One Shot solutions with multiple pruning rates

For every combination 5 models were trained, pruned and fine-tuned. The error bars correspond to the standard deviation.

All of the following figures would be better on tables

**Pruning rate 0.8**

**Pruning rate 0.7**

**Pruning rate 0.6**

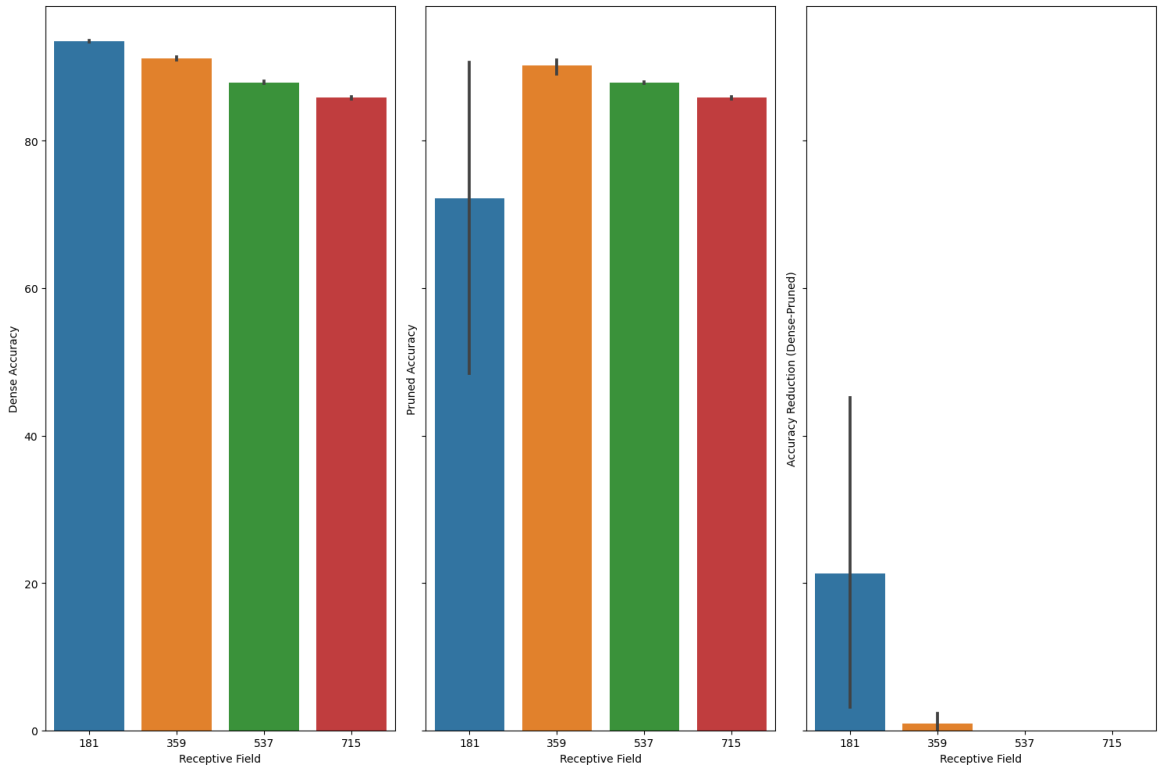
**Pruning rate 0.5**

**Fine-tuning pruned solutions**

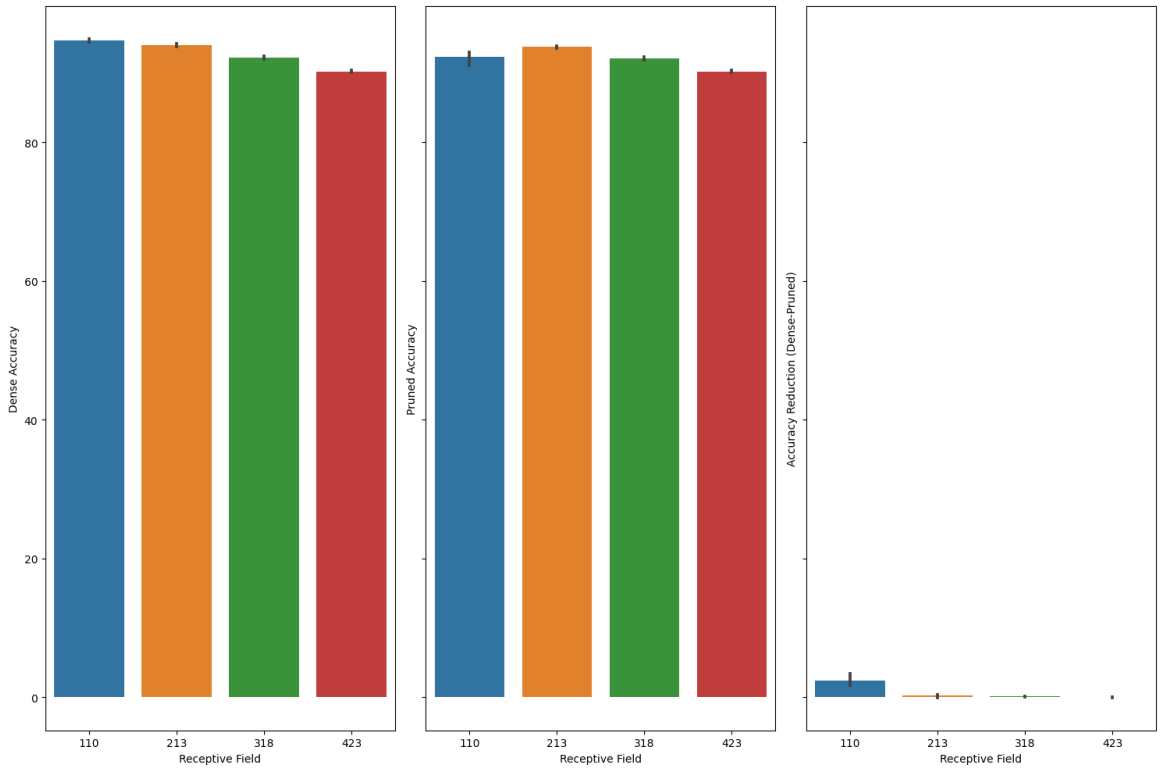
Here we fine-tuned the pruned solutions while preserving the mask for 10 epochs with the following hyper-parameters

- Initial Learning Rate: 0.0001,
- Weight Decay:5e-4
- Momentum:0,9
- Gradient clip: 0.1

All of the following figures would be better on a table

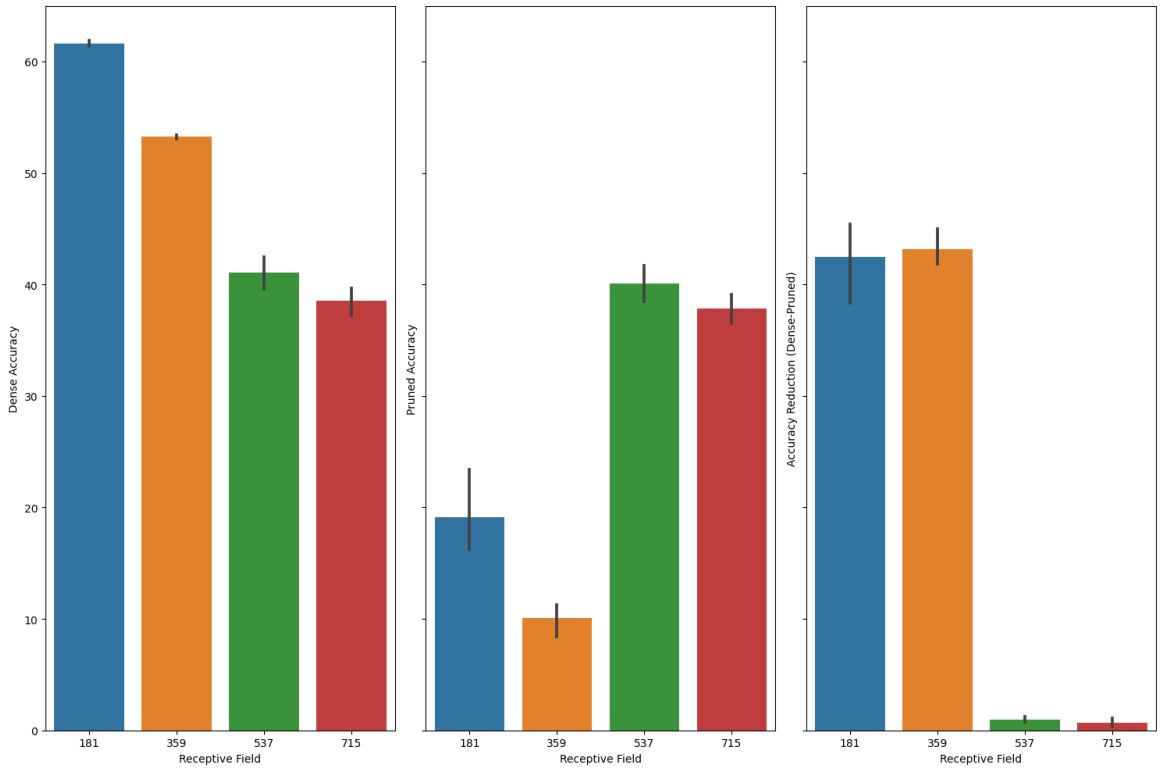



(a) VGG

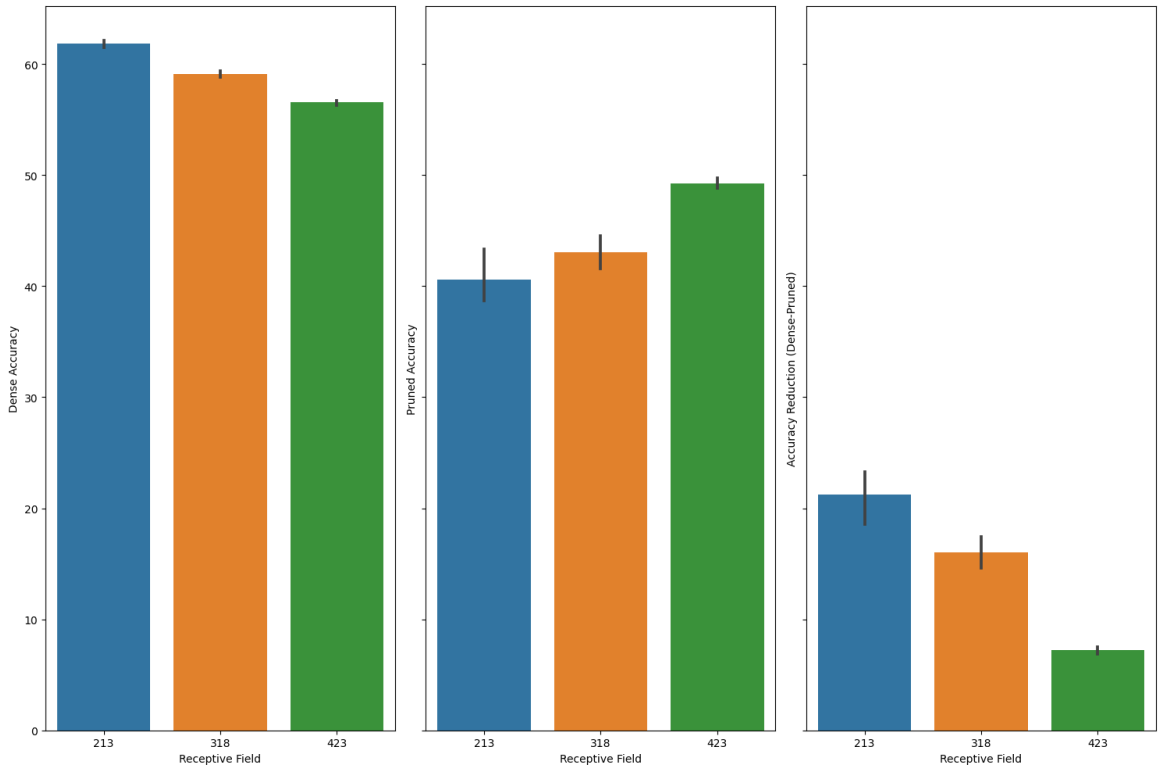



(b) ResNet-50

Figure 6: CIFAR10 pruning rate 0.8 results

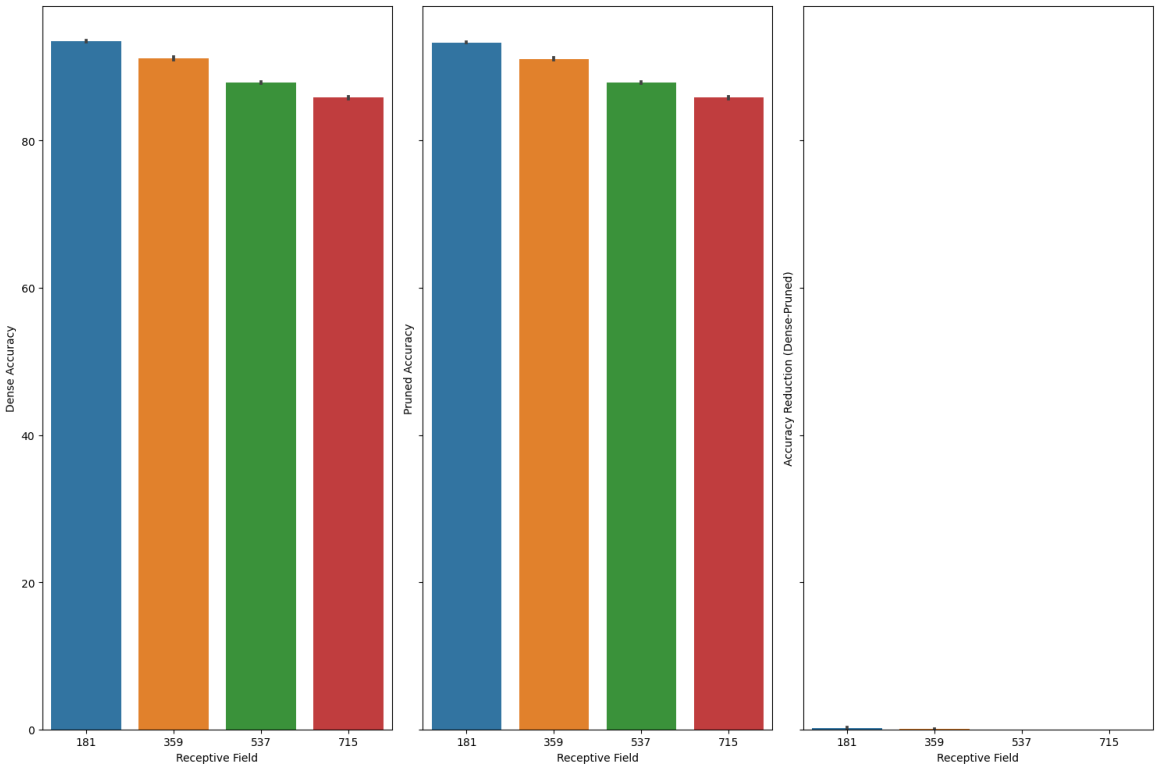



(a) VGG

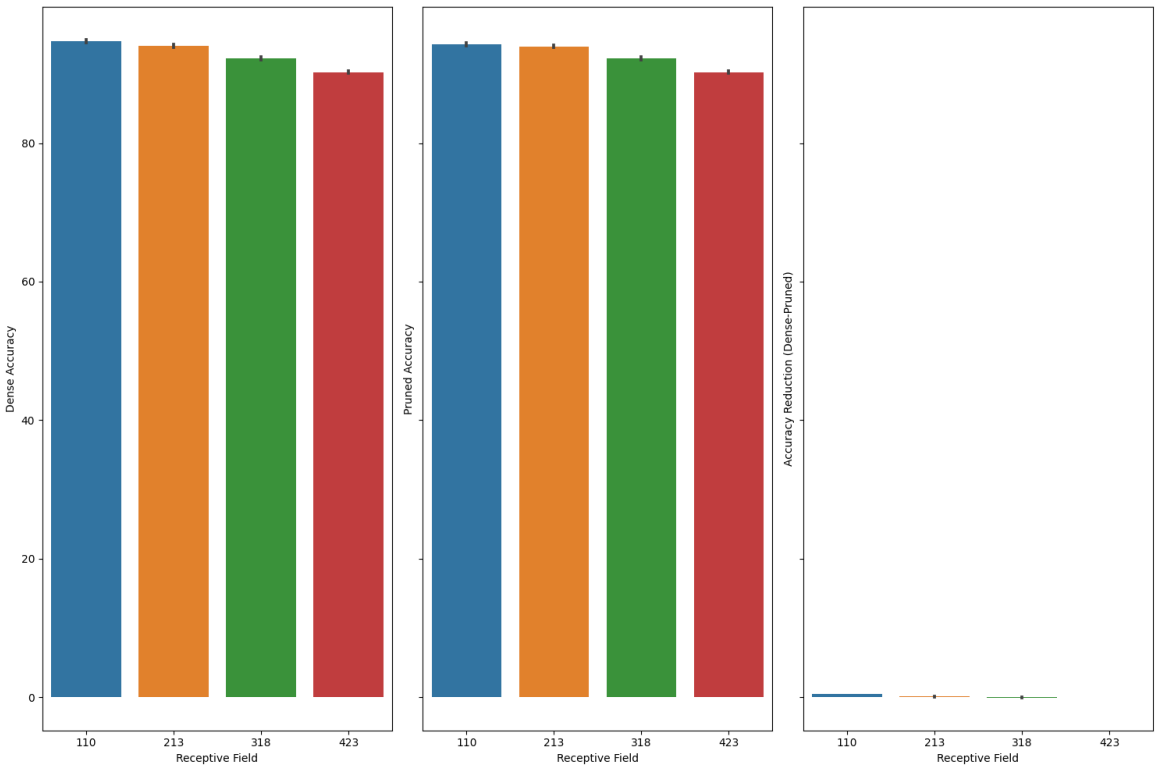



(b) ResNet-50

Figure 7: Tiny ImageNet pruning 0.8 results




(a) VGG

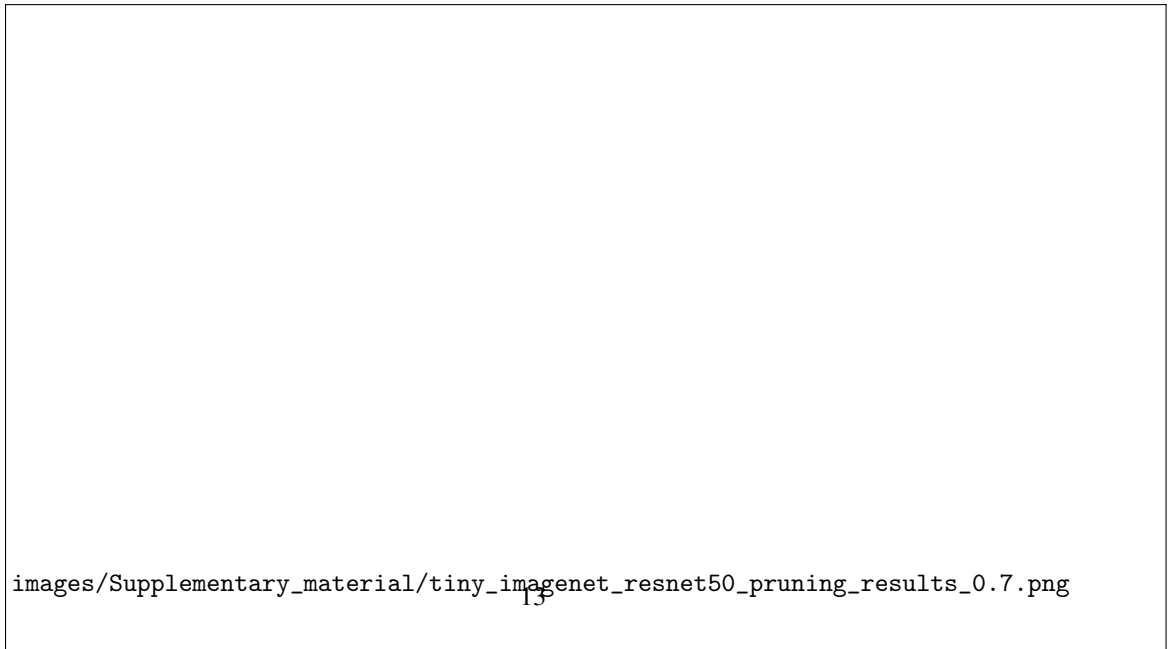


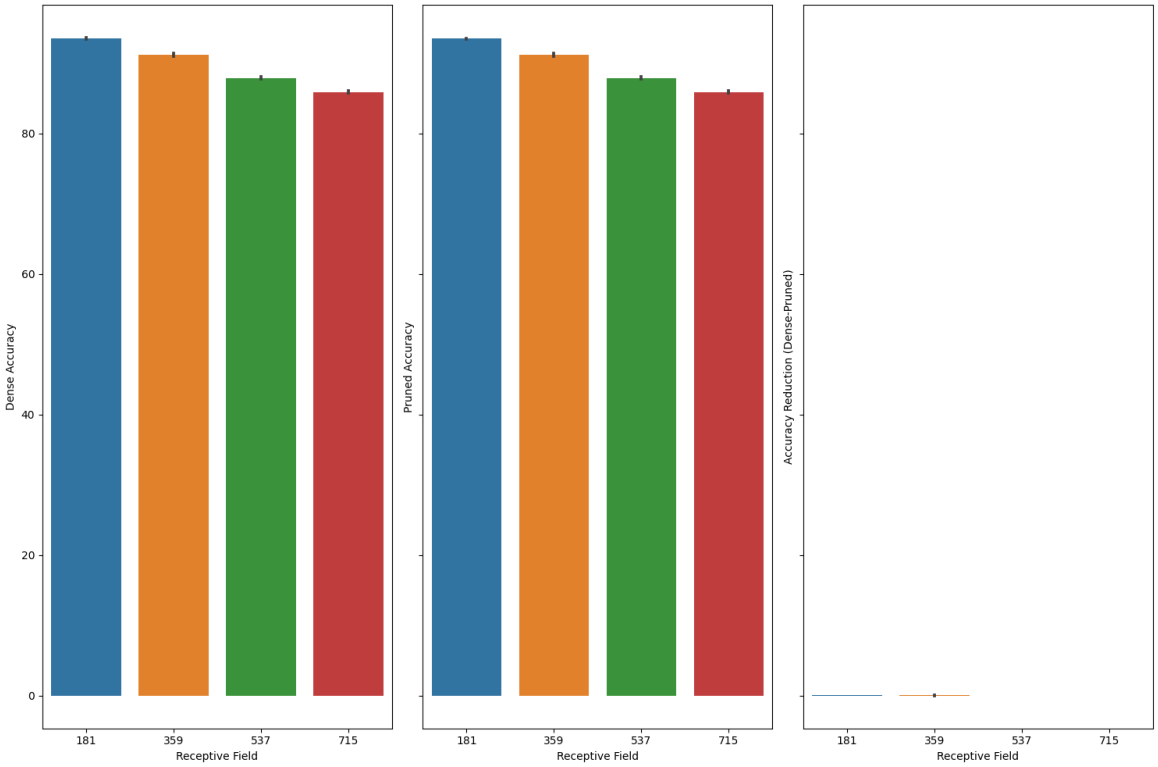

(b) ResNet-50

Figure 8: CIFAR10 0.7 results

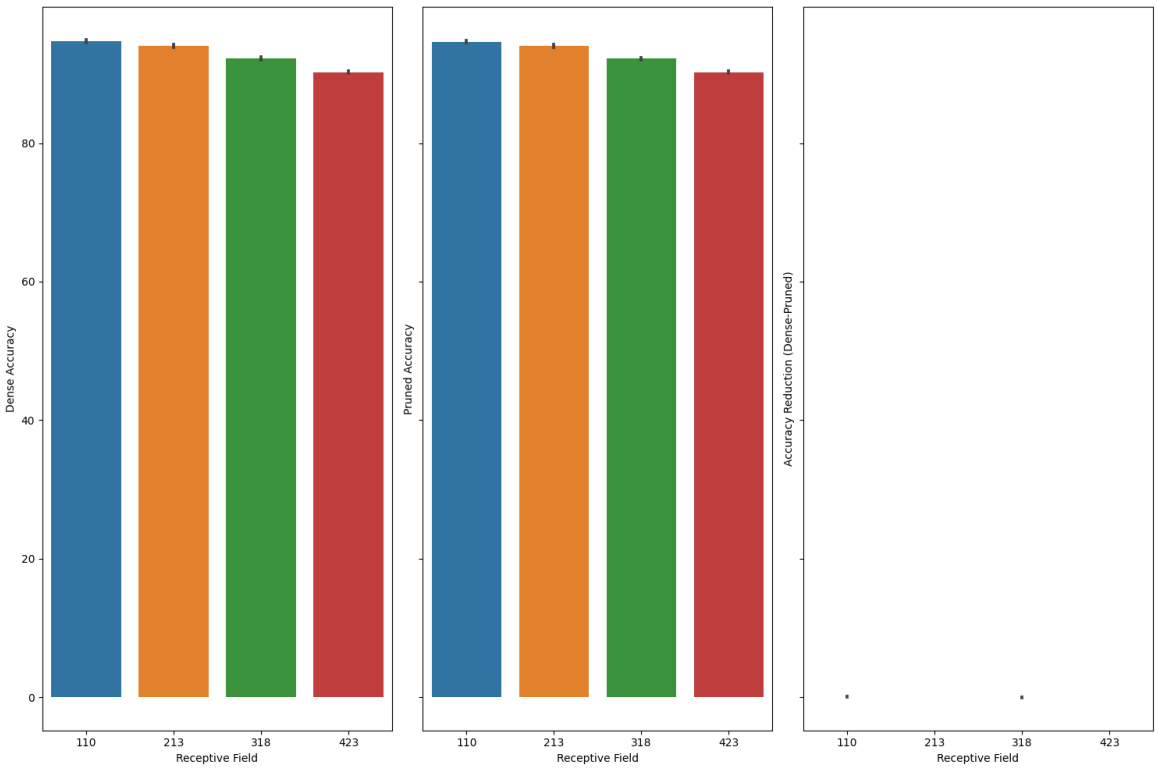

images/Supplementary_material/tiny_imagenet_vgg19_pruning_results_0.7.png

(a) VGG


images/Supplementary_material/tiny_imagenet_resnet50_pruning_results_0.7.png

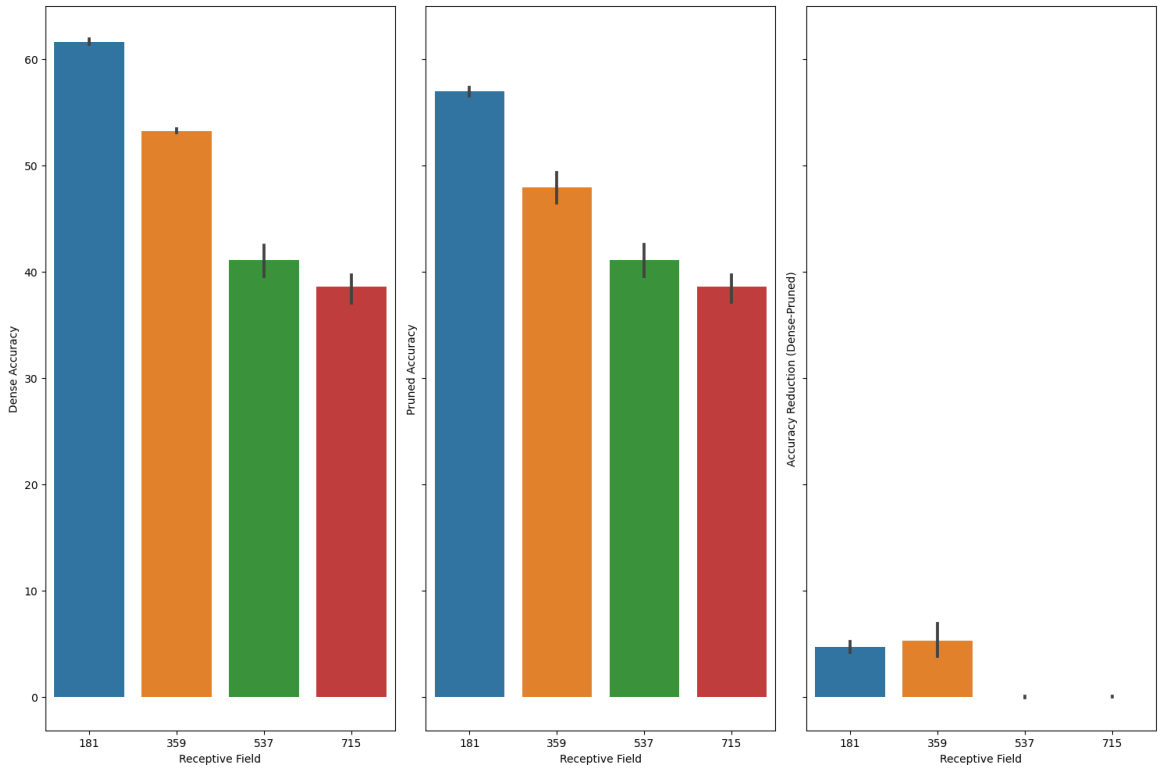



(a) VGG

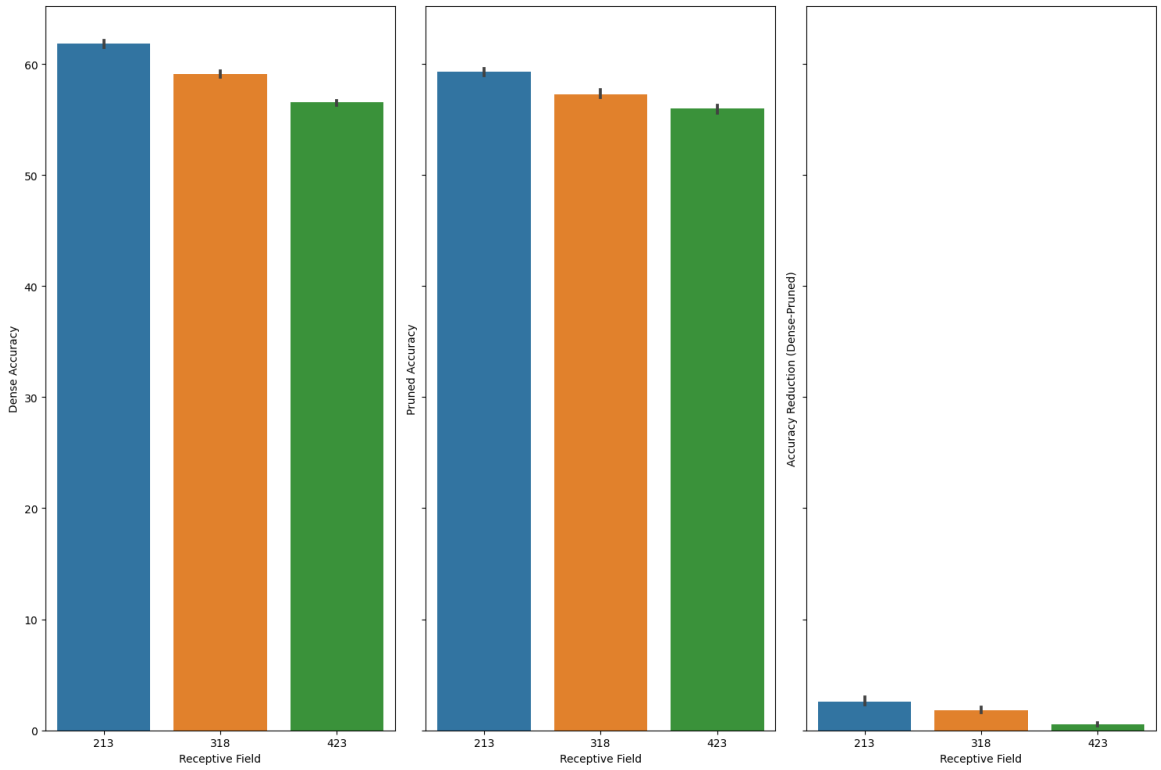



(b) ResNet-50

Figure 10: CIFAR10 0.6 results

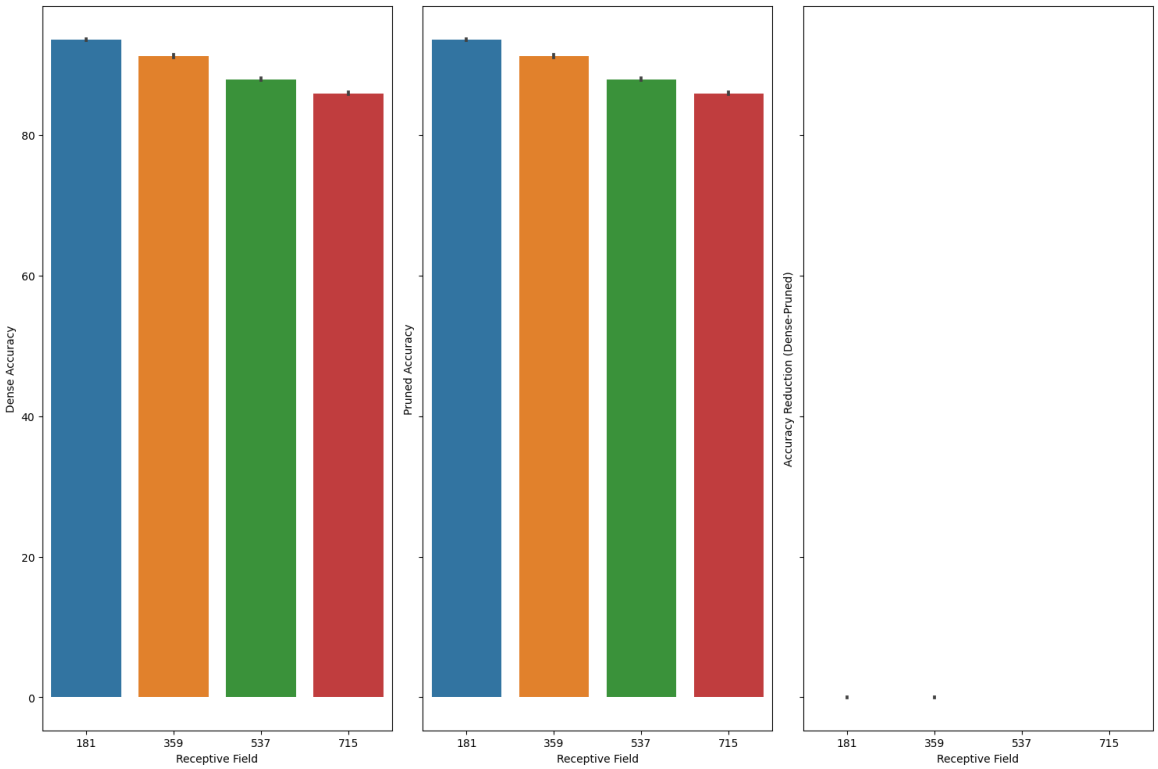



(a) VGG

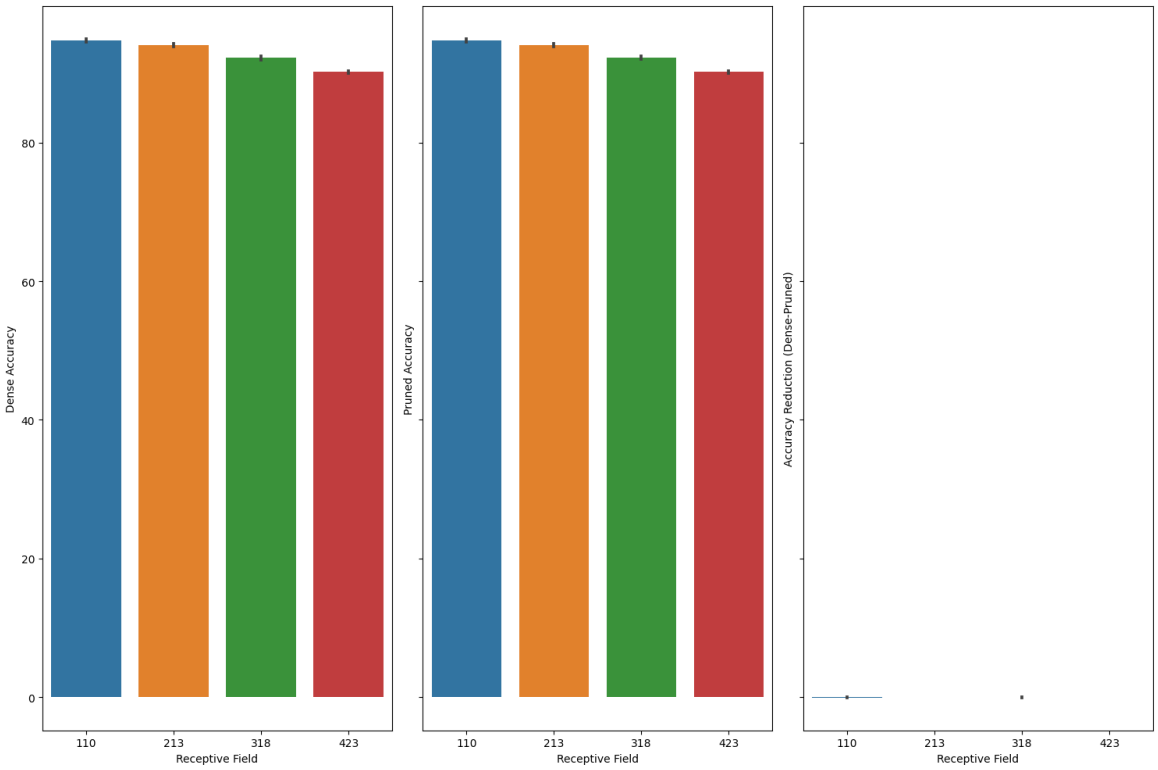



(b) ResNet-50

Figure 11: Tiny ImageNet 0.6 results




(a) VGG



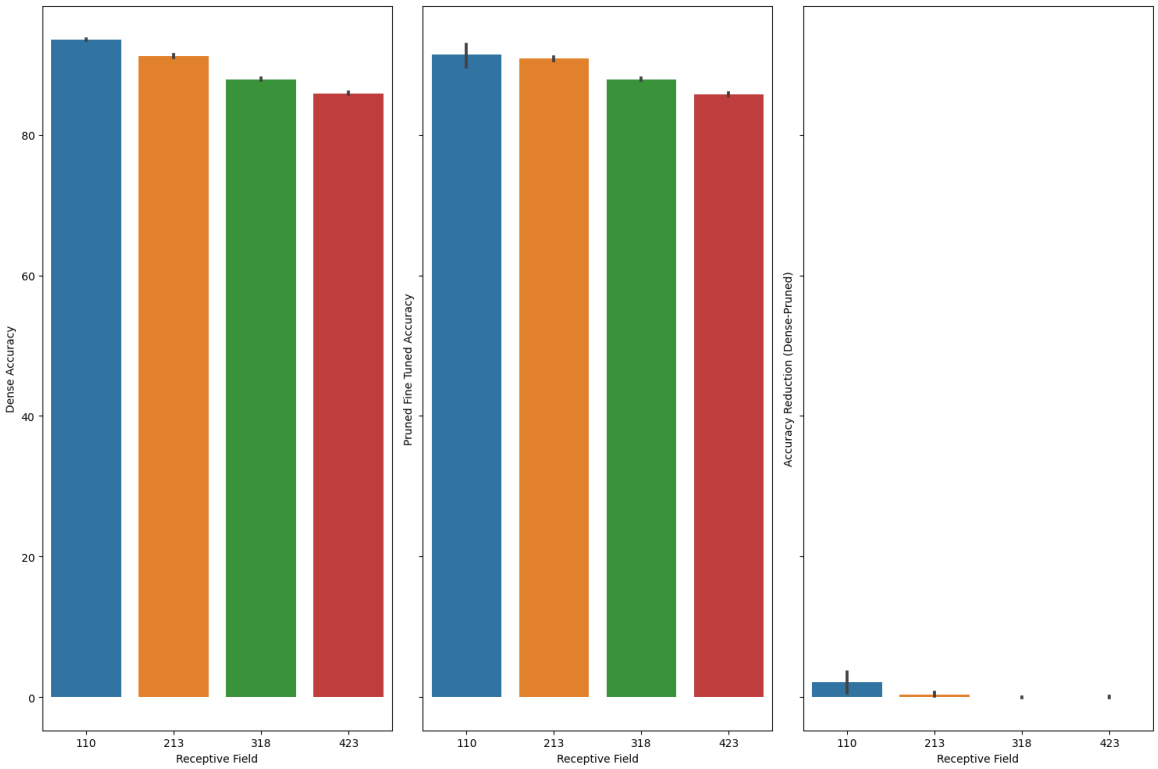

(b) ResNet-50

Figure 12: CIFAR10 0.5 results

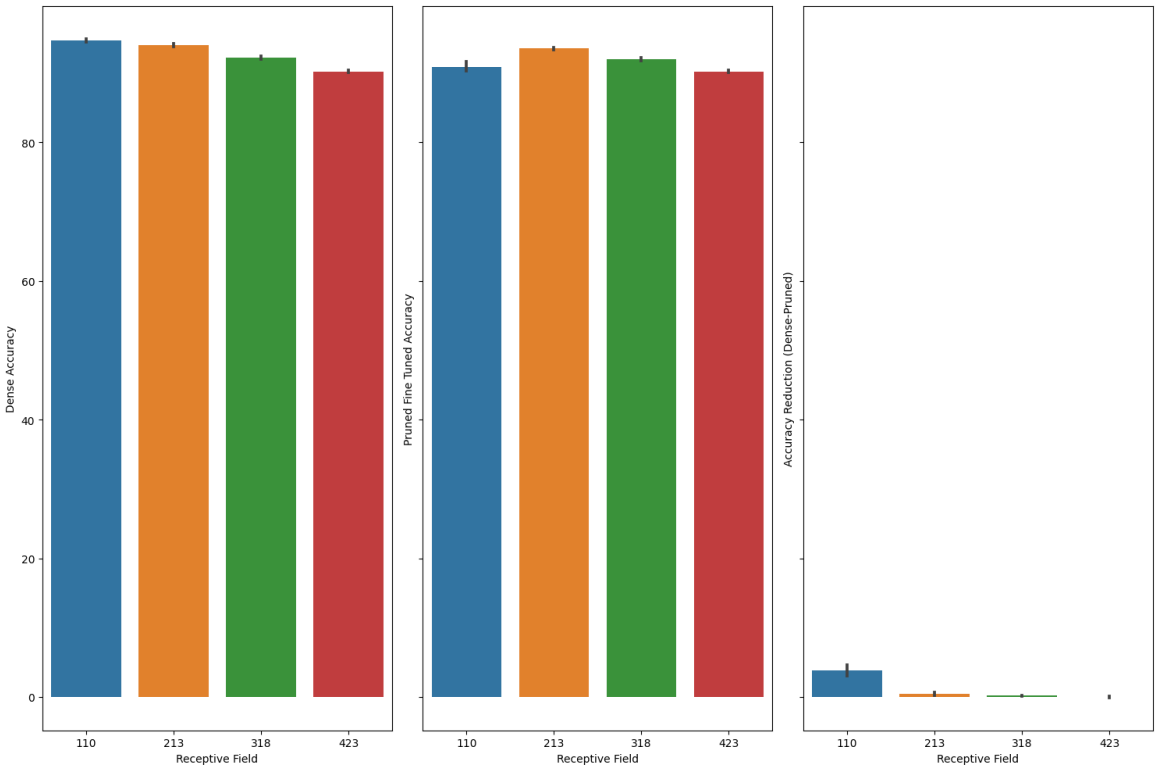

images/Supplementary_material/tiny_imagenet_vgg19_pruning_results_0.5.png

(a) VGG


images/Supplementary_material/tiny_imagenet_resnet50_pruning_results_0.5.png

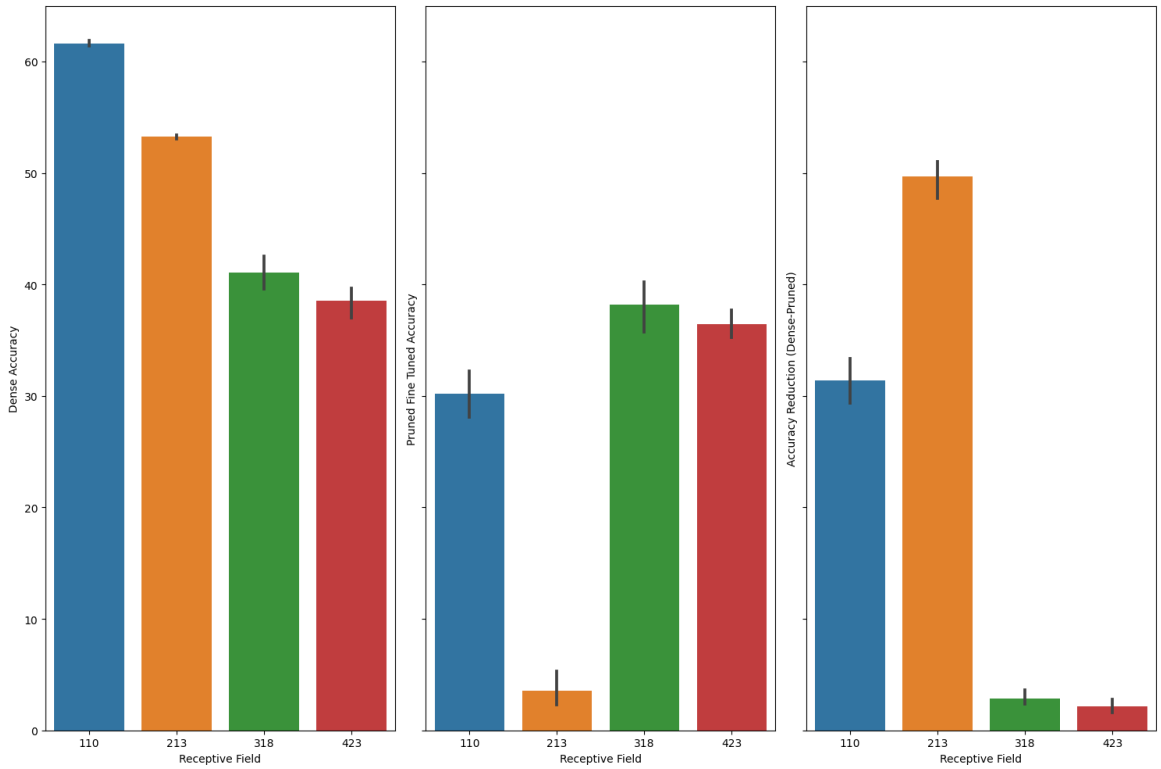



(a) VGG

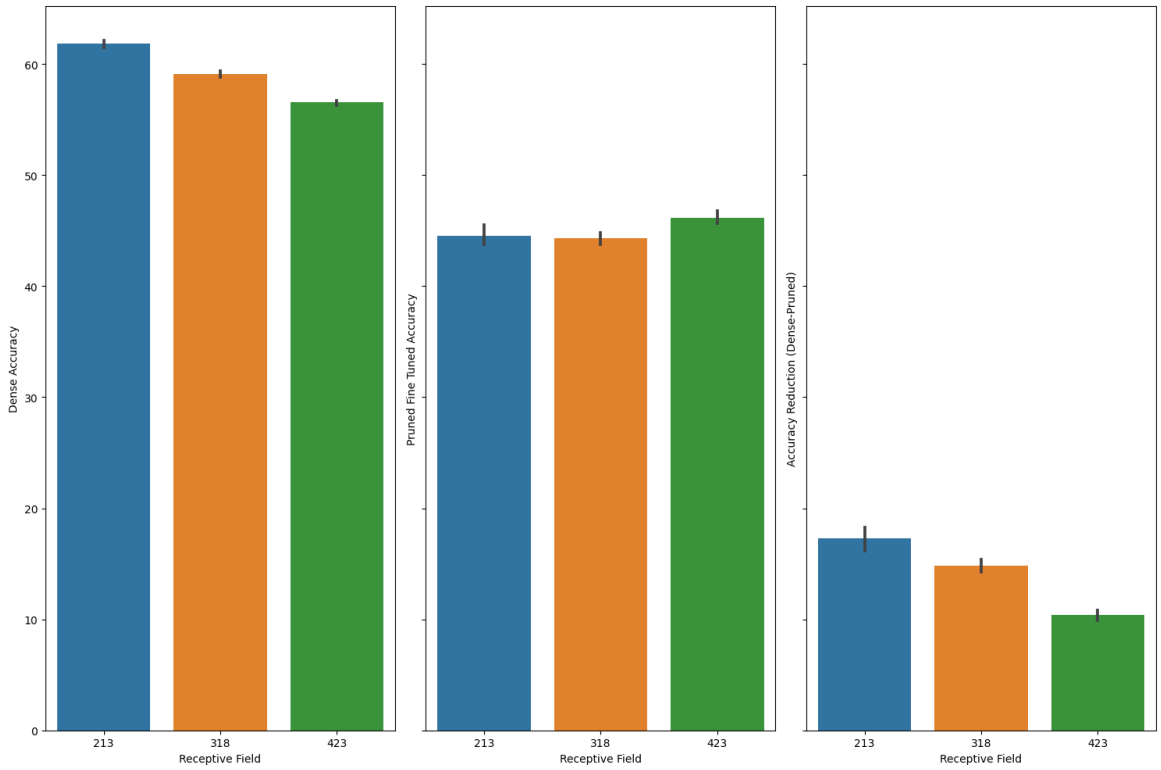



(b) ResNet-50

Figure 14: CIFAR10 Fine-tuned results




(a) VGG




(b) ResNet-50

Figure 15: Tiny ImageNet Fine-tuned results