Московский государственный технический университет им.

Н.Э. Баумана

Кафедра «Системы обработки информации и управления»

Домашнее Задание

по дисциплине

«Методы машинного обучения»

Выполнил:

студент группы ИУ5И-23М

Ван Тяньшо

Москва — 2024 г.

Задание

Домашнее задание по дисциплине направлено на анализ современных методов машинного обучения и их применение для решения практических задач. Домашнее задание включает три основных этапа:

- выбор задачи;
- теоретический этап;
- практический этап.

Этап выбора задачи предполагает анализ ресурса paperswithcode. Данный ресурс включает описание нескольких тысяч современных задач в области машинного обучения. Каждое описание задачи содержит ссылки на наиболее современные и актуальные научные статьи, предназначенные для решения задачи (список статей регулярно обновляется авторами ресурса). Каждое описание статьи содержит ссылку на репозиторий с открытым исходным кодом, реализующим представленные в статье эксперименты. На этапе выбора задачи обучающийся выбирает одну из задач машинного обучения, описание которой содержит ссылки на статьи и репозитории с исходным кодом. Теоретический этап включает проработку как минимум двух статей, относящихся к выбранной задаче. Результаты проработки обучающийся излагает в

теоретической части отчета по домашнему заданию, которая может включать:

- описание общих подходов к решению задачи;

конкретные топологии нейронных сетей, нейросетевых ансамблей или других моделей машинного обучения, предназначенных для решения задачи;

- математическое описание, алгоритмы функционирования, особенности обучения используемых для решения задачи нейронных сетей, нейросетевых ансамблей или других моделей машинного обучения;

- описание наборов данных, используемых для обучения моделей;

- оценка качества решения задачи, описание метрик качества и их значений;

- предложения обучающегося по улучшению качества решения задачи. Практический этап включает повторение экспериментов авторов статей на основе представленных авторами репозиториев с исходным кодом и возможное улучшение обучающимися полученных результатов. Результаты проработки обучающийся

излагает в практической части отчета по домашнему заданию, которая может включать:

- исходные коды программ, представленные авторами статей, результаты документирования программ обучающимися с использованием диаграмм UML, путем визуализации топологий нейронных сетей и другими способами;

- результаты выполнения программ, вычисление значений для описанных в статьях метрик качества, выводы обучающегося о воспроизводимости экспериментов авторов статей и соответствии практических экспериментов теоретическим материалам статей;

- предложения обучающегося по возможным улучшениям решения задачи, результаты практических экспериментов (исходные коды, документация) по возможному улучшению решения задачи.

Выбранная задача: «Классификация изображений»

Теоретический этап

1. Выбор задачи

Выбранная задача — классификация изображений. Классификация изображений — это основная задача компьютерного зрения, которая включает в себя присвоение входному изображению

одной из предопределенных категорий. Она имеет важное значение в различных практических приложениях, таких как автономное вождение, медицинский анализ изображений, система видеонаблюдения, приложения для смартфонов и электронная коммерция.

2. Исследуемые статьи

Статья 1: An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale

Введение:

В статье предложена новая модель для классификации изображений — Vision Transformer (ViT), основанная на архитектуре Transformers. Обычно Transformers используются для задач обработки естественного языка, но в данной статье их применяют для классификации изображений.

Основные достижения:

- Успешное применение Transformers к задаче классификации изображений, что является инновационным прорывом.
- ViT достигла результатов, сопоставимых или превосходящих традиционные сверточные нейронные сети (CNNs) на нескольких

стандартных наборах данных (например, ImageNet, CIFAR-10, CIFAR-100).

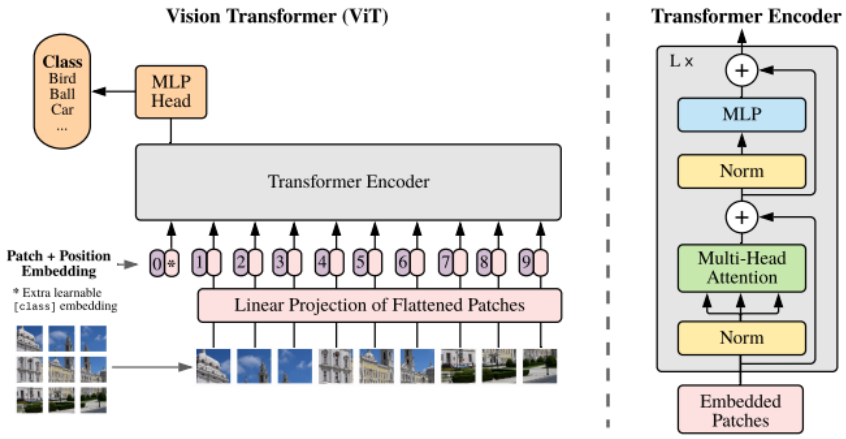


Figure 1: Model overview. We split an image into fixed-size patches, linearly embed each of them, add position embeddings, and feed the resulting sequence of vectors to a standard Transformer encoder. In order to perform classification, we use the standard approach of adding an extra learnable "classification token" to the sequence. The illustration of the Transformer encoder was inspired by Vaswani et al. (2017).

Методология:

- Разделение изображения на патчи: входное изображение делится на патчи фиксированного размера (например, 16x16 пикселей).

- Встраивание: каждый патч преобразуется в вектор фиксированной размерности с помощью линейного преобразования.

- Кодирование позиции: к каждому патчу добавляется позиционное кодирование для сохранения пространственной информации.

- Трансформер-энкодер: вектора патчей подаются на вход стандартного трансформер-энкодера.

- Классификационная голова: выход трансформер-энкодера

передается на классификационную голову для предсказания класса.

	Ours-JFT (ViT-H/14)	Ours-JFT (ViT-L/16)	Ours-I21k (ViT-L/16)	BiT-L (ResNet152x4)	Noisy Student (EfficientNet-L2)
ImageNet	**88.55** ± 0.04	87.76 ± 0.03	85.30 ± 0.02	87.54 ± 0.02	88.4/88.5*
ImageNet ReaL	**90.72** ± 0.05	90.54 ± 0.03	88.62 ± 0.05	90.54	90.55
CIFAR-10	**99.50** ± 0.06	99.42 ± 0.03	99.15 ± 0.03	99.37 ± 0.06	–
CIFAR-100	**94.55** ± 0.04	93.90 ± 0.05	93.25 ± 0.05	93.51 ± 0.08	–
Oxford-IIIT Pets	**97.56** ± 0.03	97.32 ± 0.11	94.67 ± 0.15	96.62 ± 0.23	–
Oxford Flowers-102	99.68 ± 0.02	**99.74** ± 0.00	99.61 ± 0.02	99.63 ± 0.03	–
VTAB (19 tasks)	**77.63** ± 0.23	76.28 ± 0.46	72.72 ± 0.21	76.29 ± 1.70	–
TPUv3-core-days	2.5k	0.68k	0.23k	9.9k	12.3k

Table 2: Comparison with state of the art on popular image classification benchmarks. We report mean and standard deviation of the accuracies, averaged over three fine-tuning runs. Vision Transformer models pre-trained on the JFT-300M dataset outperform ResNet-based baselines on all datasets, while taking substantially less computational resources to pre-train. ViT pre-trained on the smaller public ImageNet-21k dataset performs well too. *Slightly improved 88.5% result reported in Touvron et al. (2020).

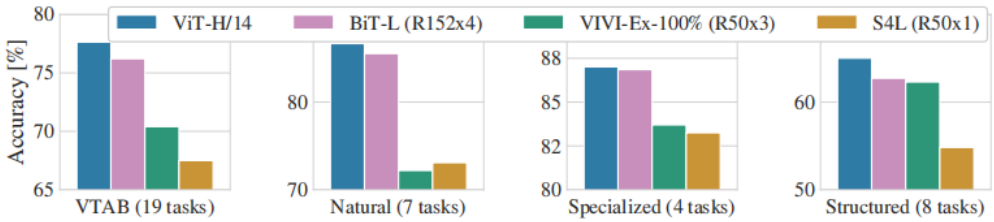


Figure 2: Breakdown of VTAB performance in *Natural*, *Specialized*, and *Structured* task groups.

Математическое описание:

Входное изображение X ∈ R^H × W × C, где H, W, C — высота, ширина и число каналов изображения соответственно.

Изображение делится на N патчей, каждый размером P × P.

Каждый патч преобразуется в вектор x_i ∈ R^P^2 · C.

Линейное преобразование: каждый патч встраивается в вектор размерности D z_i = E · x_i + p_i, где E — матрица встраивания, p_i — позиционное кодирование.

Трансформер-энкодер обрабатывает эти векторы, выход

классифицируется.

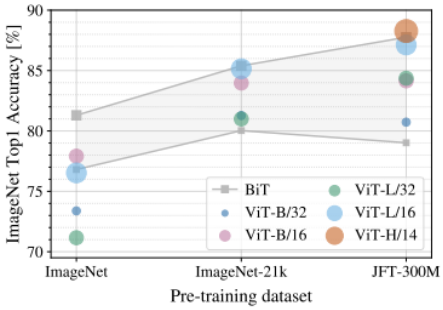


Figure 3: Transfer to ImageNet. While large ViT models perform worse than BiT ResNets (shaded area) when pre-trained on small datasets, they shine when pre-trained on larger datasets. Similarly, larger ViT variants overtake smaller ones as the dataset grows.

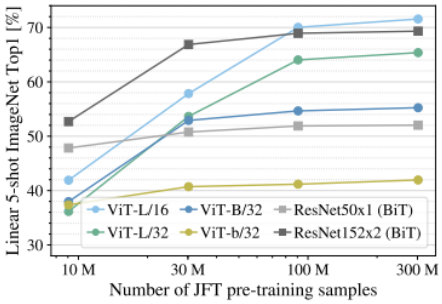
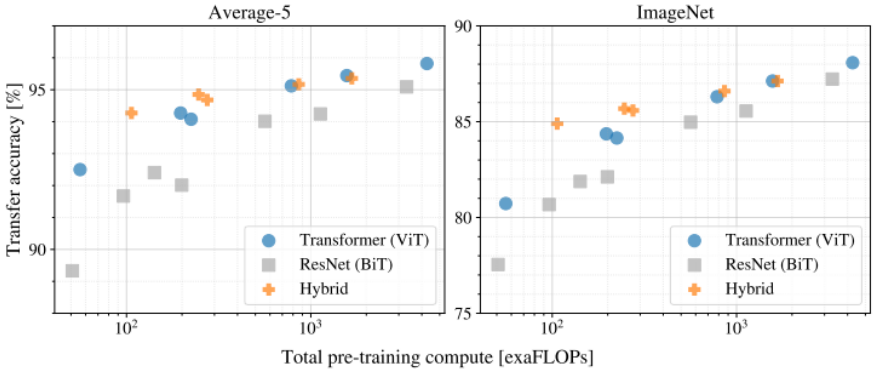


Figure 4: Linear few-shot evaluation on ImageNet versus pre-training size. ResNets perform better with smaller pre-training datasets but plateau sooner than ViT, which performs better with larger pre-training. ViT-b is ViT-B with all hidden dimensions halved.



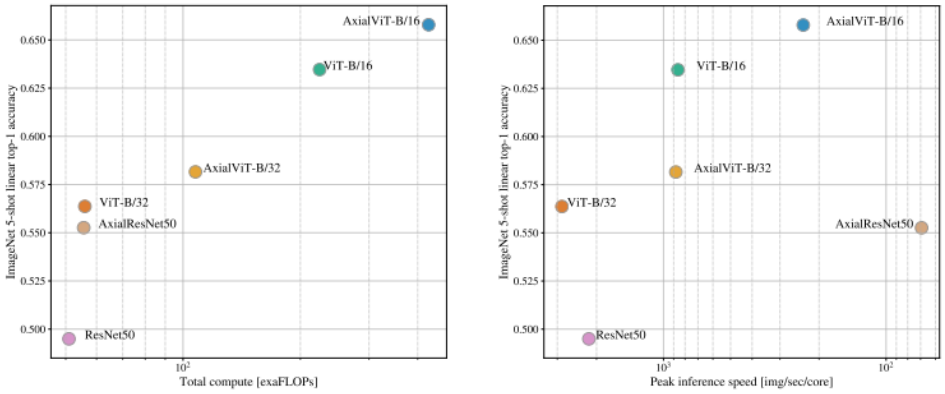


Figure 13: Performance of Axial-Attention based models, in terms of top-1 accuracy on ImageNet 5-shot linear, versus their speed in terms of number of FLOPs (**left**) and inference time (**left**).

Оценка:

- ViT достигла новых передовых результатов на ImageNet, продемонстрировав высокую способность к обобщению.

- Модель также показала отличные результаты на меньших наборах данных (например, CIFAR-10, CIFAR-100).

Статья 2: Sharpness-Aware Minimization for Efficiently Improving Generalization

Введение:

В статье предложен новый метод оптимизации — Sharpness-Aware Minimization (SAM), который стремится одновременно минимизировать значение функции потерь и крутизну (sharpness) функции потерь для улучшения способности модели к обобщению.

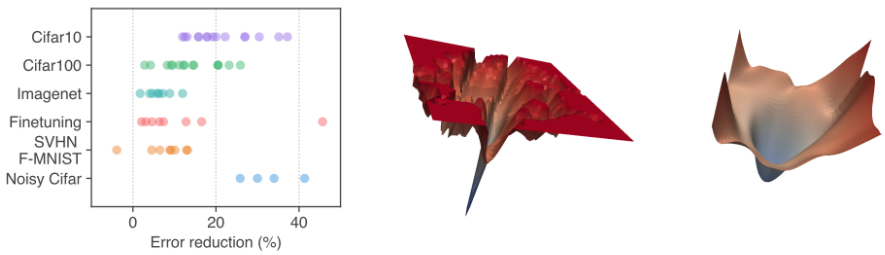


Figure 1: (left) Error rate reduction obtained by switching to SAM. Each point is a different dataset / model / data augmentation. (middle) A sharp minimum to which a ResNet trained with SGD converged. (right) A wide minimum to which the same ResNet trained with SAM converged.

Основные достижения:

- Введение алгоритма SAM, который находит гладкие минимумы

функции потерь, учитывая крутизну функции потерь во время градиентного спуска.

- Доказана эффективность SAM на нескольких стандартных наборах данных (например, CIFAR-10, CIFAR-100, ImageNet) и различных моделях.

as the base optimizer, and Figure 2 schematically illustrates a single SAM parameter update.

Input: Training set $\mathcal{S} \triangleq \cup_{i=1}^{n}\{(\boldsymbol{x}_i, \boldsymbol{y}_i)\}$, Loss function $l : \mathcal{W} \times \mathcal{X} \times \mathcal{Y} \to \mathbb{R}_+$, Batch size b, Step size $\eta > 0$, Neighborhood size $\rho > 0$.
Output: Model trained with SAM
Initialize weights $\boldsymbol{w}_0$, $t = 0$;
while *not converged* **do**
  Sample batch $\mathcal{B} = \{(\boldsymbol{x}_1, \boldsymbol{y}_1), ...(\boldsymbol{x}_b, \boldsymbol{y}_b)\}$;
  Compute gradient $\nabla_{\boldsymbol{w}} L_{\mathcal{B}}(\boldsymbol{w})$ of the batch's training loss;
  Compute $\hat{\boldsymbol{\epsilon}}(\boldsymbol{w})$ per equation 2;
  Compute gradient approximation for the SAM objective (equation 3): $\boldsymbol{g} = \nabla_w L_{\mathcal{B}}(\boldsymbol{w})|_{\boldsymbol{w}+\hat{\boldsymbol{\epsilon}}(\boldsymbol{w})}$;
  Update weights: $\boldsymbol{w}_{t+1} = \boldsymbol{w}_t - \eta \boldsymbol{g}$;
  $t = t + 1$;
end
return $\boldsymbol{w}_t$

Algorithm 1: SAM algorithm

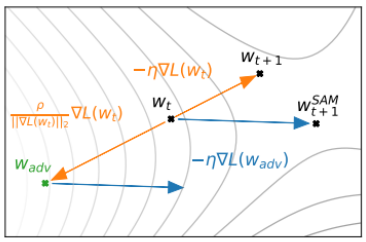


Figure 2: Schematic of the SAM parameter update.

Методология:

- Функция потерь: определяется новая функция потерь, включающая меру крутизны функции потерь.

- Градиентный спуск: используется градиентный спуск для оптимизации этой функции потерь.

- Реализация алгоритма:

1. Вычисляется стандартный градиент ∇L(w).

2. Вычисляется возмущение ε = ρ · ∇L(w) / ∇L(w), где ρ — гиперпараметр.

3. Вычисляется градиент функции потерь с учетом возмущения ∇L(w + ε).

4. Обновление весов w ← w - η · ∇L(w + ε), где η — скорость обучения.

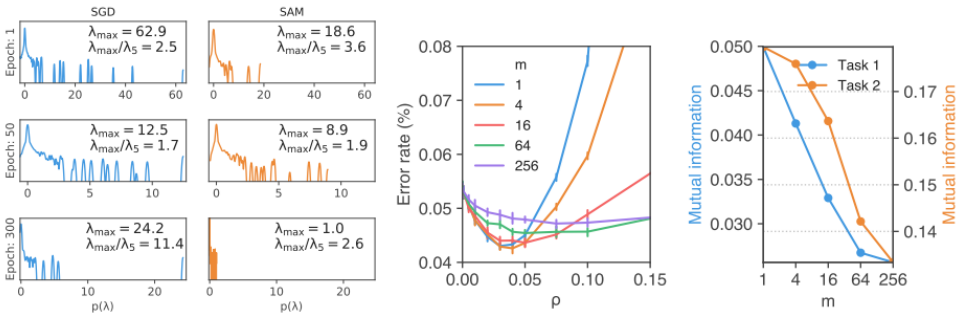


Figure 3: (left) Evolution of the spectrum of the Hessian during training of a model with standard SGD (lefthand column) or SAM (righthand column). (middle) Test error as a function of ρ for different values of m. (right) Predictive power of m-sharpness for the generalization gap, for different values of m (higher means the sharpness measure is more correlated with actual generalization gap).

Математическое описание:

Стандартный процесс обучения минимизирует тренировочную функцию потерь L_S(w), в то время как SAM минимизирует возмущенную функцию потерь L_{SAM}(w) = max_{||ε|| ≤ ρ} L_S(w + ε).

Приближенно, шаг обновления градиента становится ∇L_{SAM}(w)

$\approx \nabla L(w + \varepsilon)$.

Dataset	Efficientnet-b7 + SAM (optimal)	Efficientnet-b7 + SAM ($\rho = 0.05$)	Efficientnet-b7
FGVC_Aircraft	6.80	7.06	8.15
Flowers	0.63	0.81	1.16
Oxford_IIIT_Pets	3.97	4.15	4.24
Stanford_Cars	5.18	5.57	5.94
CIFAR-10	0.88	0.88	0.95
CIFAR-100	7.44	7.56	7.68
Birdsnap	13.64	13.64	14.30
Food101	7.02	7.06	7.17

Table 10: Results for the finetuning experiments, using $\rho = 0.05$ for all datasets.

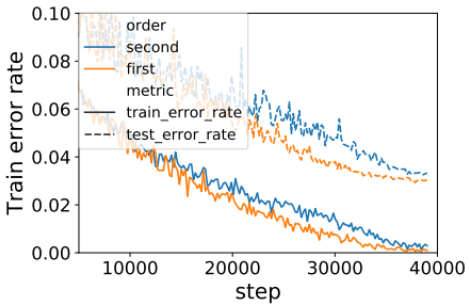


Figure 4: Training and test error for the first and second order version of the algorithm.

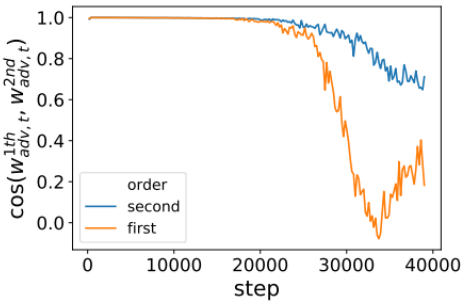


Figure 5: Cosine similarity between the first and second order updates.

Оценка:

- SAM значительно улучшил обобщающую способность моделей на различных наборах данных и моделях.

- Метод прост в реализации и может быть интегрирован в

существующие алгоритмы оптимизации.

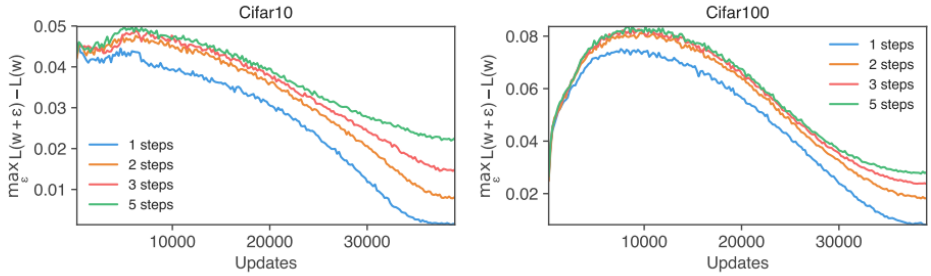


Figure 7: Evolution of $\max_{\epsilon} L(\boldsymbol{w}+\boldsymbol{\epsilon}) - L(\boldsymbol{w})$ vs. training step, for different numbers of inner projected gradient steps.

3. Сравнение методов

Архитектура моделей:

- В первой статье предложена новая модель классификации изображений ViT на основе трансформеров, тогда как во второй статье представлен новый метод оптимизации SAM.

- ViT использует разбиение изображений на патчи и обработку с помощью трансформера, в то время как SAM улучшает процесс оптимизации моделей.

Методы оптимизации:

- Первая статья сосредоточена на инновациях в архитектуре модели, а вторая — на улучшении процесса оптимизации.

- SAM может быть использован с различными архитектурами моделей, включая ViT.

Область применения:

- ViT применяется к задаче классификации изображений, но ее архитектура может быть расширена для других задач компьютерного зрения.
- SAM как метод оптимизации может быть применен к различным задачам и моделям для улучшения их обобщающей способности.

4. Предложения по улучшению

Совмещение с методами увеличения данных:

- Можно использовать больше методов увеличения данных, таких как Mixup и Cutout, в процессе обучения ViT для улучшения обобщающей способности модели.
- SAM также может быть использован вместе с методами увеличения данных для дальнейшего улучшения производительности.

Оптимизация гиперпараметров:

- Оптимизировать гиперпараметры ViT и SAM, такие как скорость

обучения, размер мини-пакета и способ кодирования позиций, чтобы найти наилучшую конфигурацию.

- Использовать автоматизированные инструменты настройки гиперпараметров, такие как Optuna или Hyperopt, для ускорения процесса оптимизации.

Интеграция методов:

- Применить метод оптимизации SAM к процессу обучения модели ViT, сочетая преимущества обоих подходов, что может привести к еще лучшим результатам.
- Исследовать комбинации различных методов оптимизации, таких как совмещение SGD и Adam с SAM.

5. Практическая часть

Практическая часть выложена в Gitlab.

vit_jax.ipynb - Colab (google.com)

https://github.com/google-research/sam

6. Список использованных источников

[1] Sharpness-Aware Minimization for Efficiently Improving Generalization.ICLR.2021 Pierre Foret, Ariel Kleiner, Hossein Mobahi, Behnam Neyshabur ·

[2] Sharpness-Aware Minimization for Efficiently Improving Generalization.ICLR.2021 · Pierre Foret, Ariel Kleiner, Hossein Mobahi, Behnam Neyshabur ·

[3] K. Simonyan and A. Zisserman, “Very deep convolutional networks for large-scale image recognition,” arXiv preprint arXiv:1409.1556, 2014. 9 [4] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, “Going deeper with convolutions,” in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2015, pp. 1–9.

[5] J. Shotton, J. Winn, C. Rother, and A. Criminisi, “Textonboost for image understanding: Multi-class object recognition and segmentation by jointly modeling texture, layout, and context,” Int. Journal of Computer Vision (IJCV), January 2009.

[6] F. Perronnin, Y. Liu, J. Sánchez, and H. Poirier, “Large-scale image retrieval with compressed fisher vectors,” in Computer Vision and

Pattern Recognition (CVPR), 2010 IEEE Conference on, 2010, pp. 3384–3391.

[7] K. E. A. van de Sande, J. R. R. Uijlings, T. Gevers, and A. W. M. Smeulders, “Segmentation as selective search for object recognition,” in IEEE International Conference on Computer Vision, 2011.

[8] C. Farabet, C. Couprie, L. Najman, and Y. LeCun, “Learning hierarchical features for scene labeling,” IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 35, no. 8, pp. 1915–1929, Aug 2013.

[9] L.-C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille, “Semantic image segmentation with deep convolutional nets and fully connected crfs,” arXiv preprint arXiv:1412.7062, 2014.

[10] V. Badrinarayanan, A. Handa, and R. Cipolla, “Segnet: A deep convolutional encoder-decoder architecture for robust semantic pixel-wise labelling,” arXiv preprint arXiv:1505.07293, 2015.