Fast GraspNeXt: A Fast Self-Attention Neural Network Architecture for Multi-task Learning in Computer Vision Tasks for Robotic Grasping on the Edge

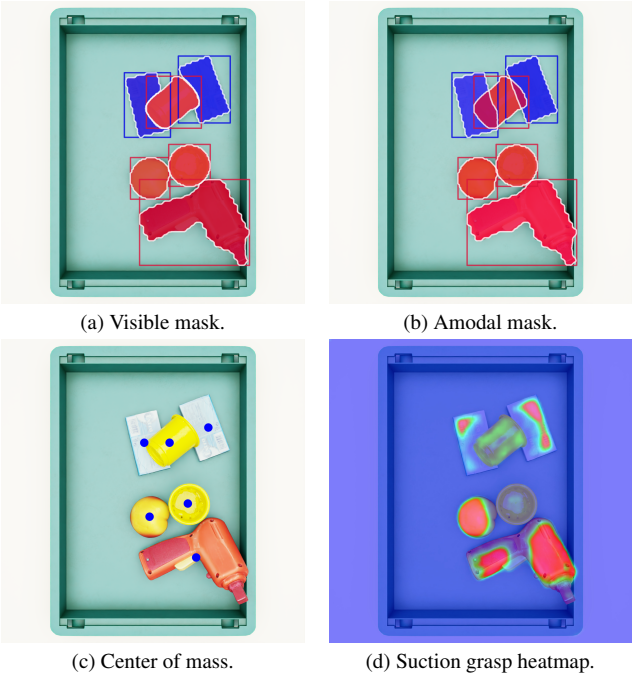
Alexander Wong[1,2,3] Yifan Wu[1] Saad Abbasi[1,3] Saeejith Nair[1]
Yuhao Chen[1] Mohammad Javad Shafiee[1,2,3]
[1] Vision and Image Processing Research Group, University of Waterloo
[2] Waterloo Artificial Intelligence Institute, Waterloo, ON
[3] DarwinAI Corp., Waterloo, ON
{a28wong, yifan.wu1, srabbasi, smnair, yuhao.chen1, mjshafiee}@uwaterloo.ca

Abstract

Multi-task learning has shown considerable promise for improving the performance of deep learning-driven vision systems for the purpose of robotic grasping. However, high architectural and computational complexity can result in poor suitability for deployment on embedded devices that are typically leveraged in robotic arms for real-world manufacturing and warehouse environments. As such, the design of highly efficient multi-task deep neural network architectures tailored for computer vision tasks for robotic grasping on the edge is highly desired for widespread adoption in manufacturing environments. Motivated by this, we propose Fast GraspNeXt, a fast self-attention neural network architecture tailored for embedded multi-task learning in computer vision tasks for robotic grasping. To build Fast GraspNeXt, we leverage a generative network architecture search strategy with a set of architectural constraints customized to achieve a strong balance between multi-task learning performance and embedded inference efficiency. Experimental results on the MetaGraspNet benchmark dataset show that the Fast GraspNeXt network design achieves the highest performance (average precision (AP), accuracy, and mean squared error (MSE)) across multiple computer vision tasks when compared to other efficient multi-task network architecture designs, while having only 17.8M parameters (about $>5\times$ smaller), 259 GFLOPs (as much as $>5\times$ lower) and as much as $>3.15\times$ faster on a NVIDIA Jetson TX2 embedded processor.

1. Introduction

Significant advances have been made in recent years to take advantage of deep neural networks for robotic grasping. In particular, multi-task learning has shown considerable promise for improving the performance of deep learning-driven vision systems for robotic grasping [3, 7, 8], where the underlying goal is to learn to perform additional tasks during the model training process. Multi-task learning has enabled not only greater precision and versatility in deep learning-driven vision systems for robotic grasping, but also enabled such systems to perform a wide range of computer vision tasks that are important for robotic grasping (see Fig. 1 for example tasks that need to be performed



(a) Visible mask. (b) Amodal mask. (c) Center of mass. (d) Suction grasp heatmap.

Figure 1. Example multi-task outputs from Fast GraspNeXt. (a) and (b) Detected occluded objects are shown in blue and non-occluded objects are shown in red. (c) The detected center of mass of each object is shown in blue. (d) Applicability of suction grasp is labelled from high to low in red, green, and blue as a heatmap.

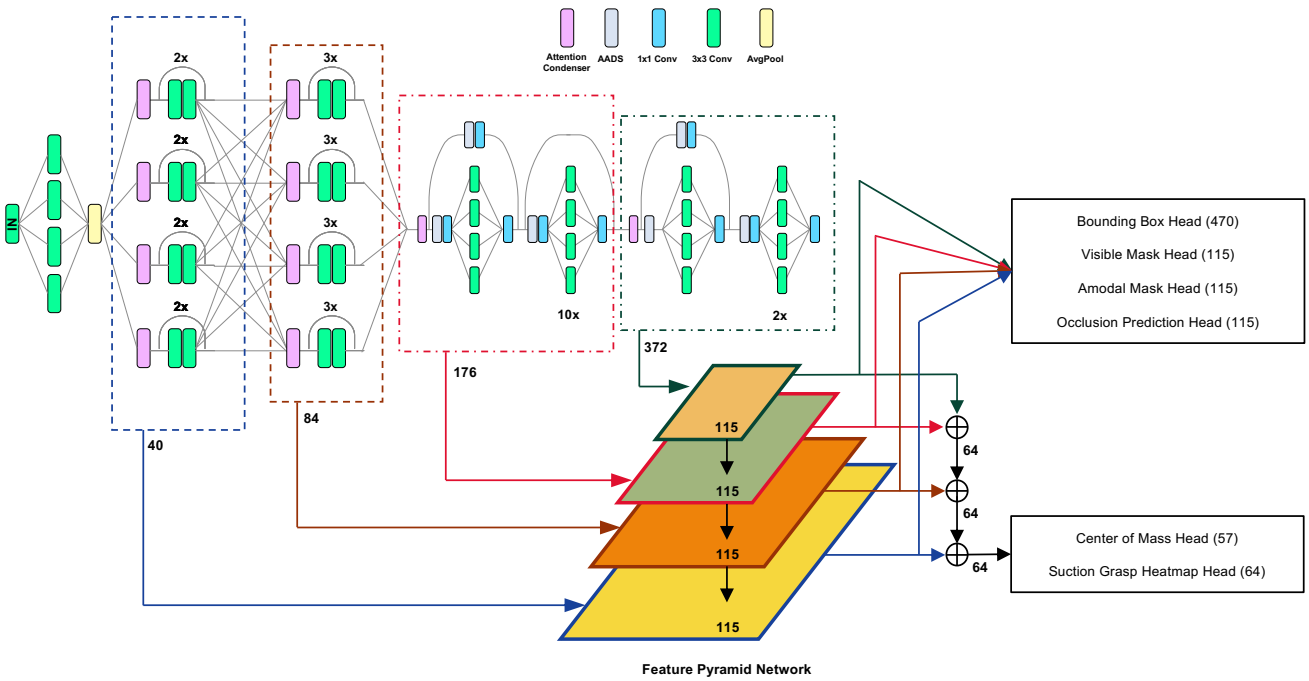


Figure 2. Overall network architecture design for Fast GraspNeXt, which possess a self-attention neural network architecture with highly optimized macroarchitecture and microarchitecture designs for all components. Fast GraspNeXt consists of a generated self-attention backbone architecture feeding into a generated feature pyramid network architecture followed by generated head network architecture designs for multi-task learning. The numbers in brackets are channel sizes of the feature maps in the heads.

by such deep learning-driven vision systems for robotic grasping such as visible object mask detection, amodal object detection [1], center of mass prediction, and suction grasp heatmap generation [2]). However, while multi-task learning can greatly improve the performance of computer vision tasks for robotic grasping, high architectural and computational complexity can limit operational use in real-world manufacturing and warehouse environments on embedded devices.

Motivated to address these challenges with embedded deployment for robotic grasping in real-world manufacturing and supply chain environments, we leverage a generative network architecture search strategy with a set of architectural design constraints defined to achieve a strong balance between multi-task learning performance and embedded operational efficiency. The result of this generative network architecture search approach is Fast GraspNeXt, a fast self-attention neural network architecture tailored specifically for multi-task learning in robotic grasping under embedded scenarios.

The paper is organized as follows. Section 2 describes the methodology behind the creation of the proposed Fast GraspNeXt via generative network architecture search, as well as a description of the resulting deep neural network architecture. Section 3 describes the dataset used in this study, the training and testing setup, as well as the experimental results and complexity comparisons.

2. Methods

2.1. Generative Network Architecture Search

In this paper, we take a generative network architecture search approach to creating the optimal multi-task deep neural network architecture for Fast GraspNeXt. More specifically, we leveraged the concept of generative synthesis [13], an iterative method that generates highly tailored architectural designs that satisfy given requirements and constraints (e.g., model performance targets). Generative synthesis can be formulated as a constrained optimization problem:

$$\mathcal{G} = \max_{\mathcal{G}} \mathcal{U}(\mathcal{G}(s)) \quad \text{subject to} \quad 1_r(G(s)) = 1, \quad \forall \in \mathcal{S}. \tag{1}$$

where the underlying objective is to learn an expression $\mathcal{G}(\cdot)$ that, given seeds $\{s|s \in S\}$, can generate network architectures $\{N_s|s \in S\}$ that maximizes a universal performance metric U (e.g., [11]) while adhering to operational constraints set by the indicator function $1_r(\cdot)$. This constrained optimization is solved iteratively through a collaboration between a generator G and an inquisitor I which inspects the generated network architectures and guides the

generator to improve its generation performance towards operational requirements (see [13] for details).

To build Fast GraspNeXt, we enforce essential design constraints through $1_r(\cdot)$ in Eq. 1 to achieve the desired balance between i) accuracy, ii) architectural complexity, and iii) computational complexity to yield high-performance, compact, and low-footprint neural network architectures such as:

1. Encouraging the implementation of anti-aliased downsampling (AADS) [14] to enhance network stability and robustness.

2. Encouraging the use of attention condensers [12], which are highly efficient self-attention mechanisms designed to learn condensed embeddings characterizing joint local and cross-channel activation relationships for selective attention. They have been shown to improve representational performance while improving efficiency at the same time.

3. Enforce a FLOPs requirement of less than 300B FLOPs and an accuracy requirement of no lower AP across all assessable tasks than a ResNet-50 variant of the multi-task network for robotic grasping (which we call ResNet-GraspNeXt) by 0.5%.

2.2. Network Architecture

The resulting Fast GraspNeXt network architecture design is shown in Fig. 2. It possesses a self-attention neural network architecture with highly optimized macroarchitecture and microarchitecture designs for all its components. The network architecture adheres to the constraints we imposed, with the generated backbone architecture feeding into a generated feature pyramid network architecture design followed by generated head network architecture designs for predicting the multi-task outputs: i) amodal object bounding boxes, ii) visible object masks, iii) amodal object masks, iv) occlusion predictions, v) object center of mass, vi) and suction grasp heatmap.

More specifically, the multi-scale features from the generated backbone architecture are provided as input directly to each level of the generated feature pyramid network architecture, followed by the generated bounding box head, visible mask head, amodal mask head and occlusion prediction head. Each level of the feature pyramid network are also upsampled to reach the same scale and summed as input for the center of mass head and suction grasp heatmap head.

The multi-task training loss, denoted as L_{mt}, used to train Fast GraspNeXt is a weighted combination of task-specific losses and can be expressed by

$$L_{mt} = l_{rpn} + \lambda_1 l_{abox} + \lambda_2 l_{segm\_v} + \lambda_3 l_{segm\_a} + \lambda_4 l_{occ} + \lambda_5 l_{com} + \lambda_6 l_{suc}, \quad (2)$$

where $\lambda_1, \lambda_2, \ldots, \lambda_6$ denote task-specific weight coefficients used to balance the contribution of individual task-specific losses. The individual task-specific losses are defined as follows:

- l_{rpn}: Region Proposal Network loss [9]
- l_{abox}: Amodal bounding box prediction loss [1]
- l_{segm_v}: Visible mask segmentation loss [1]
- l_{segm_a}: Amodal mask segmentation loss [1]
- l_{occ}: Occlusion classification loss [1]
- l_{com}: Center of mass heatmap prediction loss implemented with the modified focal loss proposed by CenterNet [15]
- l_{suc}: Suction grasp heatmap prediction loss implemented with pixel-wise averaged mean squared error (MSE) loss

.

It can be observed that the architecture design is highly heterogeneous and columnar for high architectural and computational efficiency. It can also be observed that the architecture design possesses attention condensers at different stages of the architecture for improved attentional efficacy and efficiency. Furthermore, the architecture design possesses AADS at strategic locations for greater robustness. Finally, it can be observed that the macroarchitecture for each task-specific head is unique, thus tailored around the specific balance between accuracy and efficiency for each individual task. As such, these characteristics make the Fast GraspNeXt architecture design well-suited for high-performance yet highly efficient multi-task robotic grasp applications on the edge.

3. Experiments

3.1. Dataset

We evaluate the performance of the proposed Fast GraspNeXt on the MetaGraspNet [4] benchmark dataset to explore the efficacy. This large-scale robotic grasping benchmark dataset contains 217k images across 5884 scenes featuring 82 different objects. We use 60%, 20%, and 20% of the scenes for training, validation, and testing respectively. Average precision (AP) evaluation was conducted for amodal object bounding box, visible object mask, amodal object mask, and object center of mass. Occlusion accuracy evaluation was conducted to evaluate occlusion predictions, while mean squared error (MSE) evaluation was conducted to evaluate suction grasp heatmap predictions. Our experiments use the class agnostic labels which put all objects into one class category, so that it can be readily deployed in industrial scenarios with novel, unseen items.

Table 1. Summary of quantitative performance results on MetaGraspNet dataset and network complexity.

Model	Inf. Time (ms)	Amodal Bbox AP	Visible Mask AP	Amodal Mask AP	Occlusion Accuracy	Center of Mass AP	Heatmap MSE	Parameters (M)	FLOPs (B)
ResNet-GraspNeXt	3501	85.0%	84.9%	84.1%	77.2%	75.3%	0.0113	92.1	1314
EfficientNet-GraspNeXt	2972	84.6%	85.0%	83.8%	81.7%	82.6%	0.0189	72.0	1183
MobileNet-GraspNeXt	2712	84.3%	84.6%	83.7%	80.7%	81.2%	0.0104	70.9	1189
Fast GraspNeXt	**1106**	**87.9%**	**85.4%**	**85.0%**	**85.1%**	**84.6%**	**0.0095**	**17.8**	**259**

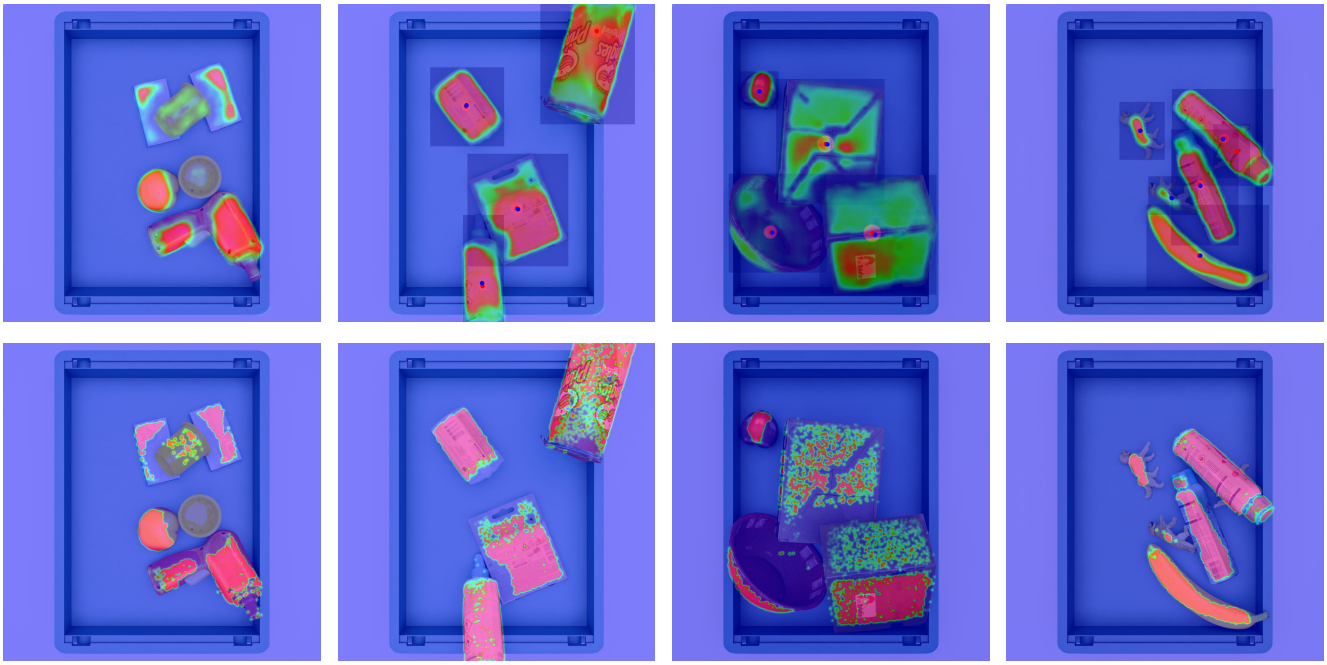


Figure 3. (top) Predicted suction grasp heatmaps produced by the proposed Fast GraspNeXt. (bottom) Example ground truth suction grasp heatmaps.

3.2. Training and Testing Setup

In addition to the proposed Fast GraspNeXt, we evaluated the performance of efficient multi-task network designs leveraging ResNet-50 [5], EfficientNet-B0 [10], and MobileNetV3-Large [6] as backbones paired with our multi-task network architecture design but without utilizing the generative network architecture search strategy. Both EfficientNet and MobileNetV3 are widely-used, state-of-the-art efficient backbones, making them well-suited for this comparison. Those network architectures are designated as ResNet-GraspNeXt, EfficientNet-GraspNeXt, and MobileNet-GraspNeXt, respectively.

For training, we use a base learning rate of 0.03, SGD optimizer with momentum of 0.9, and weight decay of 0.0001 for all experiments. Learning rate step decay are performed at 67% and 92% of the total epochs with gamma of 0.1. All network architectures are trained with the full image size of 1200×1200 pixels with batch size of 2. Empirical results found that the above training strategy yielded the best performance for all tested architectures.

Inference time evaluations are executed with batch size of 1 to reflect the robotic grasping environment which prioritise lowest possible inference latency instead of potential speed benefit of batched inference. We evaluate the inference time on the NVIDIA Jetson TX2 embedded processor with 8 GB of memory, which is widely used for embedded robotics applications in manufacturing and warehouse scenarios.

3.3. Results and Analysis

Tab. 1 shows the quantitative performance results and model complexity of the proposed Fast GraspNeXt compared to ResNet-GraspNeXt, EfficientNet-GraspNeXt, and MobileNet-GraspNeXt. We can observe that leveraging state-of-the-art efficient backbone architectures EfficientNet-B0 and MobileNetV3-Large enables noticeably faster inference time and lower architectural complexity when compared to leveraging ResNet-50 but results in

a noticeable drops in amodal bbox AP and amodal mask AP performance. In contrast, the proposed Fast GraspNeXt is >3.15×, >2.68×, and >2.45× faster on the Jetson TX2 embedded processor compared to ResNet-GraspNeXt, EfficientNet-GraspNeXt, and MobileNet-GraspNeXt, respectively, while improves the performance across all test tasks. Specifically, Fast GraspNeXt improves the amodal bbox AP, visible mask AP, amodal mask AP, occlusion accuracy, center of mass AP, and averaged heatmap MSE by 2.9%, 0.4%, 0.6%, 3.4%, 2.0%, and 8.7% respectively compared to the second best results.

In terms of architectural complexity, Fast GraspNeXt is 5.2× smaller then ResNet-GraspNeXt which has the second best amodal bbox AP and amodal mask AP, 4× smaller then EfficientNet-GraspNeXt which has the second best visible mask AP and center of mass AP, and 4× smaller then MobileNet-GraspNeXt. In terms of computational complexity, Fast GraspNeXt is 5.1×, 4.6×, and 4.6× lower FLOPs than ResNet-GraspNeXt, EfficientNet-GraspNeXt, and MobileNet-GraspNeXt respectively. Example ground truth suction grasp heatmaps along with the predicted suction grasp heatmaps produced by proposed Fast GraspNeXt are shown in Fig. 3.

As such, the above experimental results demonstrated that the proposed Fast GraspNeXt achieves significantly lower architectural complexity and computational complexity while possessing improved AP across test tasks compared to designs based on state-of-the-art efficient architectures. Furthermore, these experiments demonstrated that Fast GraspNeXt achieves significantly faster inference time on the NVIDIA Jetson TX2 embedded processor, making it well-suited for robotic grasping on embedded devices in real-world manufacturing environments. Future work involves exploring this generative approach to network architecture search for other embedded robotics applications in manufacturing and warehouse scenarios.

Acknowledgements

This work was supported by the National Research Council Canada (NRC) and German Federal Ministry for Economic Affairs and Climate Action (BMWK) under grant 01MJ21007B.

References

[1] Seunghyeok Back et al. Unseen object amodal instance segmentation via hierarchical occlusion modeling. In *IEEE Int. Conf. Robot. Automat. (ICRA)*, pages 5085–5092, 2022. 2, 3

[2] Hanwen Cao, Hao-Shu Fang, Wenhai Liu, and Cewu Lu. Suctionnet-1billion: A large-scale benchmark for suction grasping. *IEEE Robotics and Automation Letters*, 6(4):8718–8725, 2021. 2

[3] Shengqi Duan, Guohui Tian, Zhongli Wang, Shaopeng Liu, and Chenrui Feng. A semantic robotic grasping framework based on multi-task learning in stacking scenes. *Engineering Applications of Artificial Intelligence*, 121:106059, 2023. 1

[4] Maximilian Gilles, Yuhao Chen, Tim Robin Winter, E Zhixuan Zeng, and Alexander Wong. Metagraspnet: A large-scale benchmark dataset for scene-aware ambidextrous bin picking via physics-based metaverse synthesis. In *2022 IEEE 18th International Conference on Automation Science and Engineering (CASE)*, pages 220–227, 2022. 3

[5] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016. 4

[6] Andrew Howard, Mark Sandler, Grace Chu, Liang-Chieh Chen, Bo Chen, Mingxing Tan, Weijun Wang, Yukun Zhu, Ruoming Pang, Vijay Vasudevan, et al. Searching for mobilenetv3. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 1314–1324, 2019. 4

[7] J. Mahler et al. Dex-net 3.0: Computing robust vacuum suction grasp targets in point clouds using a new analytic model and deep learning. In *IEEE Int. Conf. Robot. Automat. (ICRA)*, pages 5620–5627, 2018. 1

[8] William Prew, Toby Breckon, Magnus Bordewich, and Ulrik Beierholm. Improving robotic grasping on monocular images via multi-task learning and positional loss. In *2020 25th International Conference on Pattern Recognition (ICPR)*, pages 9843–9850. IEEE, 2021. 1

[9] S. Ren, K. He, R. Girshick, and J. Sun. Faster R-CNN: Towards real-time object detection with region proposal networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 36(6):1137–1149, June 2016. 3

[10] Mingxing Tan and Quoc Le. Efficientnet: Rethinking model scaling for convolutional neural networks. In *International conference on machine learning*, pages 6105–6114, 2019. 4

[11] Alexander Wong. Netscore: towards universal metrics for large-scale performance analysis of deep neural networks for practical on-device edge usage. In *Image Analysis and Recognition: 16th International Conference, ICIAR 2019, Waterloo, ON, Canada, August 27–29, 2019, Proceedings, Part II*, pages 15–26. Springer, 2019. 2

[12] Alexander Wong, Mahmoud Famouri, Maya Pavlova, and Siddharth Surana. Tinyspeech: Attention condensers for deep speech recognition neural networks on edge devices. *arXiv preprint arXiv:2008.04245*, 2020. 3

[13] Alexander Wong, Mohammad Javad Shafiee, Brendan Chwyl, and Francis Li. Ferminets: Learning generative machines to generate efficient neural networks via generative synthesis. *arXiv preprint arXiv:1809.05989*, 2018. 2, 3

[14] Richard Zhang. Making convolutional networks shift-invariant again. In *ICML*, 2019. 3

[15] X. Zhou, D. Wang, and Philipp Krähenbühl. Objects as points. *arXiv preprint arXiv:1904.07850*, April 2019. 3