View Meta-Reviews

Paper ID
68

Paper Title
RNNPool: Efficient Non-linear Pooling for RAM Constrained Inference

META-REVIEWER #2

META-REVIEW QUESTIONS

1. Please provide a meta-review for this paper that explains to both the program chairs and the authors the key positive and negative aspects of this submission. Because authors cannot see reviewer discussions, please also summarize any relevant points that can help improve the paper. Please be sure to make clear what your assessment of the pros/cons of this paper are, especially if your assessment is at odds with the overall reviewer scores. Please do not explicitly mention your recommendation in the meta-review (or you may have to edit it later).
Reviewers appreciated that the proposed method reduces on-chip memory while maintaining performance, which is important for many practical applications. The paper is generally well written.

However, there are some concerns by reviewers about the technical novelty, comparisons to similar previous work, questions about the validity of the results, and missing ablations.

8. I agree to keep the paper and supplementary materials (including code submissions and Latex source), and reviews confidential, and delete any submitted code at the end of the review cycle to comply with the confidentiality requirements.
Agreement accepted

9. I acknowledge that my meta-review accords with the ICML code of conduct (see https://icml.cc/public/CodeOfConduct).
Agreement accepted

View Author Feedback

Paper ID
68

Paper Title
RNNPool: Efficient Non-linear Pooling for RAM Constrained Inference

AUTHOR FEEDBACK QUESTIONS

1. Author Response to Reviewers Please use this space to respond to any questions raised by reviewers, or to clarify any misconceptions. Please do not include any links to external material, nor include "late-breaking" results that are not responsive to reviewer concerns. We request that you understand that this year is especially difficult for many people, and to be considerate in your response.

We thank all reviewers for constructive feedback.

R1:

1. Peak RAM calculation: For RNNPool we design a streaming inference method (LL 303-311) where all output features of the first conv are not stored. Refer to Appendix C.2 for details. This could also be used for MobileNetV2 but the first MBConv block is the bottleneck leading to no reduction in peak RAM.
2. Typo: Yes, it should be 80x80
3. Large no.of layers: Please see response on “RAM reduction” to R4. We will rephrase the claim.

R2:

1. Meta-style methods: RNNPool is a meta-style block with a focus on efficient inference. We will add more in related work.
2. Stronger structures: We make a simple/efficient choice for RNNPool to show its utility. Adding more RNNs might boost accuracy but would increase FLOPs significantly. Sec 5 shows that RNNPool models also have comparable accuracies to base models.
3. RNNPool at deeper layers: LL 243-246 presents such an experiment.
4. ImageNet: Details of ImageNet-10 are in Appendix A.1. Typical on-device models for real world applications deal with limited classes(e.g. intruder detection). ImageNet-10 is a good proxy for this task with medium res natural images. Results in Table 3 were obtained after extensive grid search to ensure strong baselines. ImageNet-1K is a harder problem with more sensitivity to hyper-parameters. It takes ~3 days on 4 GPUs to run 1 experiment, so same hyperparameters as base models were used which is not optimal. See last para of Sec 5.1 for discussion on this. Due to large compute costs we will add more experiments in next revisions.
5. Fair experimentation: Please refer to the rebuttal on “Pooling Experiment” to R4.

R3:

1. Internals of RNNPool: Yes, RNNPool has learnable parameters in both RNNs. Pooling is an aggregation operation & RNNs are non-linear summarization techniques. Using them for pooling results in shallower networks with similar expressivity & lesser RAM/compute. See Sec-3.2, 3.3 & 4.1 for more insights. We need 2 RNNs as the first summarizes each row/column into a vector while the other accumulates the first level of summaries into an output vector.
2. Face Detection models: We agree they are inspired by S3FD but are modified with RNNPool, which is the focus: we can use RNNPool with several architectures without sacrificing accuracy.

R4:

Before addressing individual points, we would like to address the key concerns R4 has with synthetic/pooling experiments:

Synthetic:

* This was only meant to provide intuition into what RNNPool is doing, comparison with standard baselines is done on 3 different applications (Sec 5).
* See Sec 3.2 & Appendix B.1 for motivation and more details. They show that RNNPool can solve basic vision tasks (edge/shape detection etc.) which might seem counter-intuitive.

Pooling:

* We never claim that the number of layers is the key metric. In all tables, we compare various methods on

Accuracy, RAM, FLOPs.
* Comparison with pooling operators is to show that if we pool aggressively (using standard operators), accuracy drops so much that even if the operator is more efficient, it does not matter! So we first show that standard pooling does not suffice for our purpose. We then show how RNNPool, when combined with standard models, leads to comparable accuracy on multiple tasks but with much smaller RAM/FLOPs requirements (Sec 4,5).
Individual points:
1. Pooling operator: We never claim that they are expensive. See RL 11-22.
2. RAM & FLOPs: Appendix C has details on compute & RAM calculations. We do not claim that 2 RNNs have smaller memory & compute footprint compared to standard pooling but rather with the equivalent portion of the network replaced. Table 1 has FLOPs comparison for the same peak RAM across methods.
3. ReNet: See Sec 2 & first 3 paras in Appendix for a deeper discussion on ReNet. ReNet block does not seem conducive to do pooling/reduce the feature map size. We are not aware of any usage of ReNet to enable vision tasks on tiny devices.
4. No.of Layers: We do not claim that pooling is the only reason for large no. of layers in CNNs. We state that RNNPool can compress feature map aggressively, leading to much shallower network with better RAM/FLOPs footprint without compromising on accuracy. We do not propose layers replaced as a metric but evaluate RNNPool on real-world metrics of peak RAM, FLOPs & model size. Since we use patches, computation of each patch can be parallelized effectively. See Sec 3.3 & Table 1 rows 4 vs 5 for comparison with a strided conv layer.
5. RAM reduction: The intermediate outputs for calculation of a conv layer/RNNPool contribute much less to the volume of stored activations compared to output/input activation map making them the usual peak memory bottlenecks (LL 315 – RL 297 & Appendix C). For the same downsampling, a series of conv blocks could have an intermediate layer with higher RAM utilization than the input/output, making it the bottleneck. RNNPool removes dependence on such RAM hungry intermediate layers.

---

2. Confidential Comments for Meta-Reviewer If needed, you may use this space to raise any issues confidentially to the meta-reviewer. Please use this space sparingly. Please remember that meta-reviewers cannot see author identities, so please do not include any potentially de-anonymizing information.
We will like to point out that R4's review has multiple false claims and has asked us to present something when it is already given in Sections 4,5.
Claim 1: the concept of "layer" is arbitrary and...cannot be used as a metric…"
Ans: In none of our tables, we use layers as the metric, we use Accuracy, RAM, Flops as the key metric.
Claim 2: "details on how memory and FLOPs are computed are not provided"
Ans: We provide details in Section 5, para 3 and in appendix C titled "Details about Compute and Peak RAM Calculation"
Claim 3: " I also recommend to compare against well-known architectures…"
Ans: We do provide numbers against standard architectures on multiple vision tasks. E.g. MobileNets, DenseNet, ResNet, EagleEye, Faceboxes, EXTD, LFFD.

We understand that the review load is high, but claiming to be "confident in evaluation of the paper" when the reviewer didn't even look at key sections in the paper is really unfortunate and hopefully you can take this into consideration.

---

3. I certify that this author response conforms to the ICML Code of Conduct (https://www.icml.cc/public/CodeOfConduct)
Agreement accepted

View Reviews

Paper ID
68

Paper Title
RNNPool: Efficient Non-linear Pooling for RAM Constrained Inference

Reviewer #1

Questions

1. Please summarize the main claim(s) of this paper in two or three sentences.
This paper proposes a novel RNN-based pooling operator, called RNNPool. High-resolution layers in CNNs particularly those with residual connections, result in high peak RAM usage, which is not affordable for resource-constrained devices. RNNPool can replace those high-resolution layers and significantly decrease peak RAM usage in inference phase while retaining comparable accuracy. Besides, the proposed RNNPool-based face detector achieves SOTA MAP within limited resources.

2. Merits of the Paper. What would be the main benefits to the machine learning community if this paper were presented at the conference? Please list at least one.
1. RNNPool lowers on-chip memory requirement and makes many architectures affordable for memory-constrained devices .
2. The RNNPool-based face detector achieves superior performance.

3. Please provide an overall evaluation for this submission.
Borderline paper, but has merits that outweigh flaws.

4. Score Justification Beyond what you've written above as "merits", what were the major considerations that led you to your overall score for this paper?
Beyond the merits, there are some issues.

1. A question about the peak RAM usage: In Chowdhery et al., 2019, the peak RAM usage is defined as follows: “As a first-order approximation, we estimate the peak memory usage as follows. For each operation (e.g., matmul, convolution, pooling), we sum the size of the input allocations and output allocation. If the neural network is a simple chain of operations with no branching, for e.g. in MobileNet V1, select the maximum of these numbers and skip the following steps. For each parallel branch in the graph, for e.g. in MobileNet V2, we need to sum the activation storage of every pair of operations between the two branches. For each pair of operations, shared inputs must be counted once (e.g., the input to a simple residual block will be used in both parallel chains of the block) and we select the maximum of these numbers.”

Table 3 shows that the peak RAM of MobileNetV2-RNNPool is 0.24MB, but as shown in Figure 2, the output feature map of the first conv in MobileNetV2-RNNPool is 112×112×32 = 0.38MB, much higher than the peak RAM. There seems to be something wrong with the calculation method of the peak RAM usage.

2. In line 41(right), “As a result, individual entries of the 28×28 output would need to be evaluated one at a time, using expensive re-computation of large parts of the 18 intermediate layers. “ “28×28” in this sentence is not mentioned in the context. Is it should be “80×80”？

Reference:
Chowdhery, A., Warden, P., Shlens, J., Howard, A., and Rhodes, R. Visual wake words dataset. arXiv preprint arXiv:1906.05721, 2019.

5. Detailed Comments for Authors Please comment on the following, as relevant: - The significance and novelty of the paper's contributions. - The paper's potential impact on the field of machine learning. -

The degree to which the paper substantiates its main claims. - Constructive criticism and feedback that could help improve the work or its presentation. - The degree to which the results in the paper are reproducible. - Missing references, presentation suggestions, and typos or grammar improvements.
- The significance and novelty of the paper's contributions.

The RNNPool is novel. But there seems to something wrong with the calculation method of peak memory usage. We can’t see the superiority of this operation.

- The paper's potential impact on the field of machine learning.

The paper aims at a very challenging problem, which is potentially to have a big impact on deep learning applications.

- The degree to which the paper substantiates its main claims.

Good.

- Constructive criticism and feedback that could help improve the work or its presentation.

I suggest the authors check the evaluation method the peak memory usage.

- The degree to which the results in the paper are reproducible.

Good. The code is offered.

- Missing references, presentation suggestions, and typos or grammar improvements.

1.In line 41(right), “As a result, individual entries of the 28×28 output would need to be evaluated one at a time, using expensive re-computation of large parts of the 18 intermediate layers. “ “28×28” in this sentence is not mentioned in the context.

2.In line 29(right), “However, due to the large number of layers and dense residual connections in these new architectures, their working- memory requirement for inference is large. “ I think the number of layers has nothing to do with the working-memory requirement for inference.

6. Please rate your expertise on the topic of this submission, picking the closest match.
I have seen talks or skimmed a few papers on this topic, and have not published in this area.

7. Please rate your confidence in your evaluation of this paper, picking the closest match.
I tried to check the important points carefully. It is unlikely, though possible, that I missed something that could affect my ratings.

8. Datasets If this paper introduces a new dataset, which of the following norms are addressed? (For ICML 2020, lack of adherence is not grounds for rejection and should not affect your score; however, we have encouraged authors to follow these suggestions.)
This paper does not introduce a new dataset (skip the remainder of this question).

12. I agree to keep the paper and supplementary materials (including code submissions and Latex source) confidential, and delete any submitted code at the end of the review cycle to comply with the confidentiality requirements.
Agreement accepted

13. I acknowledge that my review accords with the ICML code of conduct (see https://icml.cc/public/CodeOfConduct).
Agreement accepted

Reviewer #2

Questions

1. Please summarize the main claim(s) of this paper in two or three sentences.
In this paper, the authors consider a new perspective of CNN compression, namely reducing peak RAM usage besides overall network complexity. The propose method is a meta-style pooling design, directly transforming feature maps from a large size to a much smaller size. Performance evaluation is conducted on three different visual tasks.

2. Merits of the Paper. What would be the main benefits to the machine learning community if this paper were presented at the conference? Please list at least one.
+ This paper addresses a new perspective of CNN compression.
+ The proposed pooling method for reducing peak RAM usage is simple and really interesting.
+ Competitive results on some experiments.
+ Code is provided for result reproduction.

3. Please provide an overall evaluation for this submission.
Borderline paper, but the flaws may outweigh the merits.

4. Score Justification Beyond what you've written above as "merits", what were the major considerations that led you to your overall score for this paper?
The novelty of the proposed method and its contributions, paper presentation and experiments. Please see my detailed comments.

5. Detailed Comments for Authors Please comment on the following, as relevant: - The significance and novelty of the paper's contributions. - The paper's potential impact on the field of machine learning. - The degree to which the paper substantiates its main claims. - Constructive criticism and feedback that could help improve the work or its presentation. - The degree to which the results in the paper are reproducible. - Missing references, presentation suggestions, and typos or grammar improvements.
Compressing CNN is a critical research topic in the deep learning community. Unlike most of existing works which usually focus on improving total compression performance, this work particularly considers the problem of reducing peak RAM usage besides overall network complexity. The authors smartly address this problem from the perspective of reducing memory-intensive feature size via pooling design innovation. The propose method is a meta-style pooling design, directly transforming feature maps from a large size to a much smaller size, thus may no need of stacking several layers for progressive size reduction.

My questions to this paper are as follows.

- Using meta-style network for improved DNN learning is very popular in recent years, a more thorough summary and comparison of related works (on BN/attention block etc.) is necessary.

- The meta-style pooling design has two simple RNNs, how about using more powerful structures, e.g., adding more layers rather than two?

- In the experiments, the authors directly use the meta-style pooling block to early layers, how about use it to more deep layers? That is, with more aggressive feature size reduction.

- Regarding the experiments on ImageNet, it poses serious problems for fair comparison. Creating ImageNet-10 from ImageNet-1K increases the risk of unfair comparisons. (1) Such a small self-created dataset from ImageNet-1K is easy to lead over-fitting; (2) Also there is no reference results available for comparison; (3) Comparing the result differences shown in Table 3 and Table 4 clearly validates my above concerns. It can be clearly seen that with the same network (MobileNetV2 and EfficientNet-B0), the proposed method shows much worse results on ImageNet-1K than ImageNet-10, why? How about the performance of other three networks (ResNet18, DenseNet121 and GoogLeNet) on ImageNet-1K?

- For comparisons with max/average pooling and stride convolution, experimental settings are also not fair enough to some extent. With reduced network depth, these parameter free methods will definitively lead poor accuracy. The authors should provide some experiments to directly replace existing counterparts in the original networks, in this context, the meta-style pooling design having more learnable parameters may show improved accuracy?

Post rebuttal

I carefully read the rebuttal and other reviews. I really appreciate Reviewer#4's very detailed and constructive comments. First, I must admit that I did not notice "ReNet: A Recurrent Neural Network Based Alternative to Convolutional Networks" from Yoshua Bengio's team, which is available on Arxiv since 2015. I carefully read the work of ReNet, and mostly agree with R4's concerns on the novelty of this submission.

Besides, this submission also needs following improvements: (1) experimental ablations on the different choices of RNN designs, e.g., structure, depth and inserted position; (2) experiments on the ImageNet-10 are not convincing, leading to sharp result differences against those on the ImageNet-1K as can be seen from the paper.

Considering above aspects, I lower my score, and lean to reject this submission.

6. Please rate your expertise on the topic of this submission, picking the closest match.
I have published one or more papers in the narrow area of this submission.

7. Please rate your confidence in your evaluation of this paper, picking the closest match.
I tried to check the important points carefully. It is unlikely, though possible, that I missed something that could affect my ratings.

8. Datasets If this paper introduces a new dataset, which of the following norms are addressed? (For ICML 2020, lack of adherence is not grounds for rejection and should not affect your score; however, we have encouraged authors to follow these suggestions.)
The dataset has a persistent identifier such as Digital Object Identifier or Compact Identifier.

12. I agree to keep the paper and supplementary materials (including code submissions and Latex source) confidential, and delete any submitted code at the end of the review cycle to comply with the confidentiality requirements.
Agreement accepted

13. I acknowledge that my review accords with the ICML code of conduct (see https://icml.cc/public/CodeOfConduct).
Agreement accepted

Reviewer #3

Questions

1. Please summarize the main claim(s) of this paper in two or three sentences.
This paper proposes a pooling algorithm based on RNN to rapidly downsample image size. Authors showed that RNNPool layer can be succefully replace several layers of deep networks.

2. Merits of the Paper. What would be the main benefits to the machine learning community if this paper were presented at the conference? Please list at least one.
RNNPool is a novel and interesting idea. However since pooling layers usually do not have any learnable parameters, it is not clear what is the point of using RNN which is usually used for learning sequences.

3. Please provide an overall evaluation for this submission.
Borderline paper, but has merits that outweigh flaws.

4. Score Justification Beyond what you've written above as "merits", what were the major considerations that led you to your overall score for this paper?
It is an interesting idea, however it is not well justified why RNNs are a good choice for a pooling operator which does not have any learnable parameters.

5. Detailed Comments for Authors Please comment on the following, as relevant: - The significance and novelty of the paper's contributions. - The paper's potential impact on the field of machine learning. - The degree to which the paper substantiates its main claims. - Constructive criticism and feedback that could help improve the work or its presentation. - The degree to which the results in the paper are reproducible. - Missing references, presentation suggestions, and typos or grammar improvements.
The novelty of the paper lies in using RNN for pooling operators. While RNNPool is the main component of this paper, the explanation of RNNPool is very short and not sufficiently descriptive The explanation is limited to 5 lines of pseudocode (#12-17) and only one paragraph explanation (line 143-150 in the second column). it is not clear if RNNPool has any learnable parameters? why RNN is a good case for pooling? Are there any sequences in the pooling operator that makes RNN a good candidate? why a pair of RNN is used? if we have a matrix of numbers representing an image we know clearly how max and average pooling operator outputs are but RNNPool seems very ambiguous. it seems that RNNPool is just adding more ambiguity to CNNs rather than making it more explainable.

it is not clear what are the contributions of authors in the new architecture for face detection. the architecture is based on S3FD and RNNPool layer so I dod not agree that it is a NEW architecture or it is an additional contribution.

6. Please rate your expertise on the topic of this submission, picking the closest match.
I have closely read papers on this topic, and written papers in the broad area of this submission.

7. Please rate your confidence in your evaluation of this paper, picking the closest match.
I tried to check the important points carefully. It is unlikely, though possible, that I missed something that could affect my ratings.

12. I agree to keep the paper and supplementary materials (including code submissions and Latex source) confidential, and delete any submitted code at the end of the review cycle to comply with the confidentiality requirements.
Agreement accepted

13. I acknowledge that my review accords with the ICML code of conduct (see https://icml.cc/public/CodeOfConduct).
Agreement accepted

Reviewer #4

Questions

1. Please summarize the main claim(s) of this paper in two or three sentences.
The authors introduce the RNNPool layer, a recurrent layer intended as a replacement to the ubiquitous pooling layers with a smaller memory and computation footprint. This layer is based on two RNNs with shared weights that scan the input vertically and horizontally (column- and row-wise), followed by two bidirectional RNNs that scan each resulting feature map and whose output is finally concatenated and into a flattened vector.

2. Merits of the Paper. What would be the main benefits to the machine learning community if this paper were presented at the conference? Please list at least one.
The paper proposes an architecture for image processing, based on RNNs.

3. Please provide an overall evaluation for this submission.
Below the acceptance threshold, I would rather not see it at the conference.

4. Score Justification Beyond what you've written above as "merits", what were the major considerations that led you to your overall score for this paper?

The paper is generally well-written (modulo minor exceptions), but in my opinion the proposed approach lacks novelty and the claims regarding memory and computational savings are not well substantiated.

While the previous literature is well covered, I feel a proper comparison that highlights differences and advantages of the proposed method w.r.t. similar ones is missing. Specifically, the proposed method is almost identical to ReNet by Visin et Al., with the only difference being the way the RNNs are applied on the input. It is unclear to me that there is any advantage in using an RNNPool layer over a ReNet one. Either way, the novelty of the approach seems very questionable.

Furthermore, the authors seem to consider the number of layers of an architecture (which can include arbitrarily many operations) to be indicative of the memory and computation resources used, which is not the case, and details on how memory and FLOPs are computed are not provided.I am not convinced that the proposed method can be more memory and computation efficient than pooling layers, which seems to be the claim of the paper.
Finally, in my opinion the two synthetic experiments are flawed (see detailed review below) and unconvincing. For all the above reasons I believe it should be rejected.

5. Detailed Comments for Authors Please comment on the following, as relevant: - The significance and novelty of the paper's contributions. - The paper's potential impact on the field of machine learning. - The degree to which the paper substantiates its main claims. - Constructive criticism and feedback that could help improve the work or its presentation. - The degree to which the results in the paper are reproducible. - Missing references, presentation suggestions, and typos or grammar improvements.
One of the main points of the paper seems to be that pooling operations are expensive in terms of memory. I fail to see why this would be the case, and the paper doesn't support this argument.

How are memory usage and FLOPS computed? Since this is a central point to the paper, the exact formula should be provided and discussed. I hardly believe that a 2 RNNs can have a smaller memory and computational footprint than a pooling layer and I'd be curious to see the exact math.

As I mentioned earlier, while the authors mention ReNet, they fail to highlight the similarities (surprisingly many) and the differences (few). By comparing the two, it is unclear to me what is the advantage of the proposed method over ReNet, and this should be made very clear in the paper. On top of the approach not being particularly different from ReNet, the ReNet architecture also seems more convenient to me: what would be the advantage of processing the rows and cols separately with RNN1, and then their feature maps separately with RNN2, rather than processing the input in one direction bidirectionally first, and then processing the resulting feature map in the other direction bidirectionally? ReNet is more efficient (4 RNN sweeps instead of 6) and seems to be as convenient to summarize the input patch. The advantage of the proposed architecture over ReNet, if any, should be a central point of the paper.

The paper seems to build on the notion that depth in CNN architectures has the sole goal of downsampling the images (e.g. "Using RNNPool, we can rapidly down-sample images and activation maps, eliminating the need for many residual blocks in the CNN architectures"). This is incorrect, as depth improves the expressivity of the model as well by adding non-linearities to the computation.

Throughout the paper, the authors insist that the proposed RNNPoolLayer allows to downsample images by a large factor using only one layer. While this seems to be regarded as one of the main selling points of this approach, I am not convinced this is necessarily an interesting metric and furthermore I also believe this not to be true for the proposed model. Firstly, the concept of "layer" is arbitrary: an RNNPoolLayer effectively amounts to processing the input or some intermediate feature map with 6 RNNs (some of which with shared weights). Is this to be considered one layer or six then? It is always possible to define a layer as a collection of sublayers, this doesn't make it any smaller. Secondly, what really matters is the number of FLOPs and memory usage, and how parallelizable the approach is. CNNs can be implemented in a very efficient manner, at the expense of a higher memory use, while RNNs are inherently sequential. It is unclear to me that there is a computational or memory-usage advantage in using the proposed RNNPoolLayers. In particular, I strongly disagree with the argument made at lines 073-080: the example doesn't prove the point, as the same output size would be

obtainable with a CNN with the same stride and receptive field. The dimension of the output feature map should not be used as a proxy for the peak memory usage.

Synthetic experiments
- Capturing edges, etc:
I don't think this experiments is adding much value to the paper, as I can't see any reason to doubt that the architecture would have succeeded in such a task. Also, patch size and stride should be reported. Finally, the fact that adding a CNN layer before the RNNPool one makes the model much more parameter efficient seems to contradict the main thesis that the RNNPool layers have a lower memory footprint than typical CNNs.

Comparison with pooling operators:
I believe the experiment to be inconclusive. As I said before, the concept of "layer" is arbitrary and hence the number of layers cannot be used as a metric to compare architectures. Please use an objective one, such as the amount of computation, memory and parallelization, instead. For the comparison to be “apple-to-apple”, I recommend to select two architectures with similar computation or memory or both, and to be very clear on the settings in which the comparison is made. The amount of computation of the RNNPool layer is obviously much higher than a CNN + pooling layer, so the comparison is not fair. Furthermore, it is unclear what kind of architecture is used for the CNN: is it a 32x32 pooling followed by a 1x1 convolution? If that's the case, this isn't a very strong baseline to compare against. The pooling in such an architecture would be an impossibly small bottleneck that drops most of the information, and it would be impossible for the CNN to recover such information afterwards. This experiment should be rerun with baselines with similar memory/computation to the proposed network. I also recommend to compare against well-known architectures, to guarantee the models used as a baseline are sensible.

Minor:
- Figure 1 is not clear: a more informative caption would go a long way here. In the left part, why do the green arrows of RNN2 span over what I believe is their output (the green feature maps) rather than what I believe is their input (the blue feature maps)? As for the right part, it isn’t really informative and I suggest dropping it entirely because it doesn’t add to the understanding of what a RNNPoolLayer is or does. Indeed, the figure depicts what “stride” is, that is well known in the literature and not central to the proposed algorithm.
- Line 173 left: what is multi-dimensional scaling? Is it some form of PCA or dimensionality reduction? Can you add a reference and a description of the technique?
- Line 175 left, it is not immediately clear that the “(1)” refers to the caption of the figure, when talking about the dataset. Do not refer to the dataset before it’s introduced in the text please (i.e., move the “Dataset (1) consists ..” sentence at the beginning of the paragraph)
- I couldn’t understand the meaning of the last sentence of 3.2.
- The caption of Table 1 should be improved. Please describe exactly the architectures, and which layers have been replaced and how.

Thank you for your reply. After reading the rebuttal I still see a number of issues with this paper, the biggest one being that the authors misinterpreted the ReNet paper which is why they think their approach to be more novel than it actually is. In short, ReNet does not process the input row- and column-wise, but rather does so row-wise only, and then processes *the resulting feature map* column-wise. This difference is small but substantial (see below for a more detailed explanation).

Quoting the text in the appendix, these are the fundamental differences that they find between their method and the ReNet-based methods:
1) "In ReNet based methods, the RNN is used to find a pixelwise mapping from a voxel of the input activation map to that of the output map. However, in our method we are using RNNs to spatially summarize a big patch of the input activation map to a 1 × 1 voxel of the output activation map."

The same is done in e.g., ReSeg (by the same authors as ReNet), where a patch (instead of a single pixel) is processed at each step by the RNN.
Note that this is also equivalent to first processing the image with a CNN and then applying a vanilla ReNet module, so I don't see any novelty here.

2) "[...] in ReNet the hidden states of every timestep of RNN contributes to one voxel of the output, whereas in our case only the last hidden states of the traversals are taken for both row/column wise summarizations and bidirectional summarizations. A problem with using every hidden state to determine the output as in ReNet is that the earlier hidden states do not contain significant information as compared to the last one i.e. the information keeps accumulating till the last timestep"

This is unfortunately not true, the authors misinterpreted the ReNet model. Figure 5 depicts perfectly the authors' misconception: only one of the 2 bidirectional RNNs that form a ReNet module sweeps the input, not both of them.
Specifically, a ReNet module sweeps the image row-wise with a bidirectional RNN and then sweeps the resulting feature map (not the input!) column-wise with another bidirectional RNN. Each location of the first feature map is then an embedding that captures the information of a full row (since it has information coming from both directions, thanks to the bidirectional RNN), but is also specific to a single position in the input. The second RNN builds on the feature maps of the first, and processes it column-wise. Since each position in the first embedding has information on a full row, processing it column-wise in both directions allows the second RNN to have the full input as a context in each location. Effectively, each location of the output of a ReNet layer is specific to a single location in the input, but is conditioned on the whole input at the same time. It is hence incorrect to say that "the earlier hidden states do not contain significant information as compared to the last one i.e. the information keeps accumulating till the last timestep".

In conclusion, I don't see any advantage in the proposed architecture over the vanilla ReNet module. The differences are very marginal and, ultimately, ReNet is more efficient (4 RNN sweeps instead of 6) but as effective.

6. Please rate your expertise on the topic of this submission, picking the closest match.
I have published one or more papers in the narrow area of this submission.

7. Please rate your confidence in your evaluation of this paper, picking the closest match.
I am very confident in my evaluation of the paper. I read the paper very carefully and I am very familiar with related work.

8. Datasets If this paper introduces a new dataset, which of the following norms are addressed? (For ICML 2020, lack of adherence is not grounds for rejection and should not affect your score; however, we have encouraged authors to follow these suggestions.)
This paper does not introduce a new dataset (skip the remainder of this question).

12. I agree to keep the paper and supplementary materials (including code submissions and Latex source) confidential, and delete any submitted code at the end of the review cycle to comply with the confidentiality requirements.
Agreement accepted

13. I acknowledge that my review accords with the ICML code of conduct (see https://icml.cc/public/CodeOfConduct).
Agreement accepted