Refining Remote Photoplethysmography Architectures using CKA and Empirical Methods

Nathan Vance and Patrick Flynn
University of Notre Dame
{nvance1,flynn}@nd.edu

Abstract

Model architecture refinement is a challenging task in deep learning research fields such as remote photoplethysmography (rPPG). One architectural consideration, the depth of the model, can have significant consequences on the resulting performance. In rPPG models that are over-provisioned with more layers than necessary, redundancies exist, the removal of which can result in faster training and reduced computational load at inference time. With too few layers the models may exhibit sub-optimal error rates. We apply Centered Kernel Alignment (CKA) to an array of rPPG architectures of differing depths, demonstrating that shallower models do not learn the same representations as deeper models, and that after a certain depth, redundant layers are added without significantly increased functionality. An empirical study confirms how the architectural deficiencies discovered using CKA impact performance, and we show how CKA as a diagnostic can be used to refine rPPG architectures.

1. Introduction

Remote Photoplethysmography (rPPG) is a technique for inferring the pulse waveform of a subject using digital video sequences of the subject's skin (usually the face). Recent advances in rPPG have employed deep 3D convolutional neural networks (3DCNNs) with great success, achieving error rates of less than 5 BPM in challenging scenarios [17, 25].

It is important and informative to probe the architectures of rPPG models to understand their critical and redundant portions. If a model is too shallow it may underperform. However, a deep model with many redundant layers requires more computational resources to load and train than a shallower counterpart with redundancies stripped.

The fine-tuning of architecture depths may be achieved through a brute-force parameter sweep (i.e., train a wide array of architectures and select the one that minimizes errors). However, it is informative to understand why shallower and deeper models fail while others succeed, as this informs further architectural refinements.

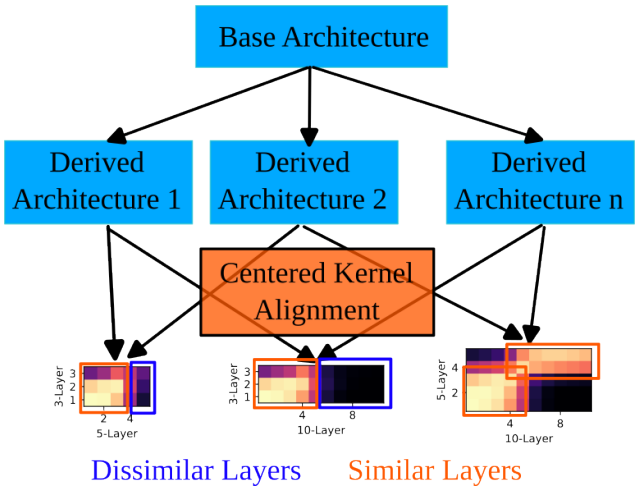


Figure 1. Overview of using CKA to inform architecture refinement by revealing similar and dissimilar layers between architectures.

In this study we make the following contributions:

- We develop an array of PhysNet-3DCNN [25] variants ranging from 2 to 15 layers in depth, as well as TS-CAN [9] variants ranging from 1 to 10 meta-layers in depth.
- We perform a Centered Kernel Alignment (CKA) [5] analysis to yield insights into network pathologies, revealing both network redundancies and also critical representations.
- We demonstrate that the findings from the CKA analysis are reflected in empirical results, arguing that such techniques should be used by the research community to refine network architectures.

Figure 1 shows an overview of our workflow. Using CKA, we compare architectures, determining which layers correspond to each other across architectures, and which

contain representations unique to that architecture.

2. Related Work

2.1. Remote Photoplethysmography

Remote Photoplethysmography (rPPG) is a technique to infer a subject's pulse waveform from video data, pioneered by Takano and Ohta in [19] and Verkruysse et al. in [23]. Poh et al. developed an early technique using blind source separation [15]. Noteworthy classical techniques include CHROM (developed by deHaan and Jeanne [4]), which was designed to be robust against motion, and POS (developed by Wang et al. [24]), which relaxes assumptions made in CHROM regarding skin tone.

Deep learning techniques for rPPG have proliferated due to their success at this task, and can be divided into two general camps: frame-difference processing architectures which compute the rPPG derivative between frames [2, 9, 11, 14, 28], and sequence processing architectures which perform rPPG over an entire clip in an end-to-end fashion [8, 17, 20, 25, 26].

Frame-difference processing architectures are among the earlier deep learning rPPG systems to be developed and have retained relevancy as lightweight alternatives to end-to-end techniques. Chen and McDuff developed DeepPhys, a 2DCNN-based architecture with two branches, one processing frame differences and the other processing raw frames [2]. Liu *et al*. expanded the capabilities of the DeepPhys architecture with TS-CAN which is capable of multi-task learning of physiological signals such as respiration and the pulse waveform [9]. Zhao *et al*. also developed this dual branch architecture by employing a 3D central difference convolution operation for noise reduction [28]. Some additional non-end-to-end architectures utilize spatial-temporal maps, including the work of Niu *et al*. which passes these maps into a ResNet-18 backbone followed by a gated recurrent unit to predict the heart rate directly [14], and Dual-GAN developed by Lu *et al*. which models both the pulse waveform and its noise distribution using two GAN modules [11]. Due to the availability of an implementation of TS-CAN in the open source rPPG-Toolbox [10], and its adoption as a state-of-the-art baseline for rPPG research, we select TS-CAN as the frame-difference architecture to which we apply our techniques.

In 2019 Yu *et al*. explored the use of end-to-end architectures in rPPG by developing PhysNet in two variants, an end-to-end variant based on 3DCNNs, and a non-end-to-end variant that uses 2DCNN feature extractors followed by a recurrent network [25]. The 3DCNN structure was explored by Lee *et al*. with Meta-rPPG, which utilizes an hourglass encoder-decoder structure which forces the network to consider the entire timeframe [8]. Tsou *et al*. additionally developed the 3DCNN architecture by using Siamese 3DCNN networks to jointly predict rPPG signals over the forehead and cheek, then combining the result into a single pulse waveform [20]. Speth *et al*. proposed RPNet which relaxes assumptions the 3DCNN PhysNet architecture makes regarding framerate by including temporal dilations [17]. In addition, video transformer networks have also been investigated for rPPG, with Yu *et al*. developing PhysFormer [26]. In this work we apply our techniques to PhysNet-3DCNN because it is an early work on which several end-to-end rPPG architectures are based.

2.2. Neural Network Similarity

The comparison of neural network representations is an important task for understanding their underlying pathology. Early pioneers in this area include Laakso and Cottrell, who base their network comparison on distance between network activations [7]. Raghu et al. developed a technique called Singular Vector Canonical Correlation Analysis (SVCCA) that allows network comparisons between different layers and architectures [16]. Morcos et al. build on SVCCA, proposing Projection Weighted CCA (PWCCA), which better differentiates between signal and noise [12]. Kornblith et al. propose using Centered Kernel Alignment (CKA) for network similarity analysis, which they find is more reliable in light of different network initializations relative to CCA-based approaches [5]. Other families of methods exist, most notably Representational Similarity Analysis (RSA), which is used heavily by the neuroscience research community [6].

Cui *et al*. provide some critique on CKA and RSA, finding that they may indicate high similarity in random networks or perform inconsistently in transfer learning, and propose modifications to resolve these issues [3]. However, our analysis in Section 3.2 verifies that CKA reveals the behavior of models as required from an effective diagnostic tool for architecture refinement.

3. Methods

3.1. Flexible Depth Models

We performed a parameter sweep over the 3DCNN based PhysNet architecture developed by Yu et al. [25], and TS-CAN developed by Liu et al. [9], by varying the depth of the networks. In this section we discuss modifications made to the published architectures to facilitate this study.

3.1.1 PhysNet-3DCNN

PhysNet-3DCNN, or simply 3DCNN, is an rPPG architecture utilizing 10 Conv3d layers operating over video frame sequences. In particular, the input video is cropped around the face and downsampled to 64×64 pixels using cubic in-

Depth	Pooling Indices	Spatial Stride
2	1	64
3	1,2	8,8
4	1,2,3	4,4,4
5	1,2,3,4	2,4,2,4
6	1,2,3,4,5	2,2,2,2,4
7	1,2,3,4,5,6	2,2,2,2,2,2
8	1,2,3,4,5,7	2,2,2,2,2,2
9	1,2,3,4,6,8	2,2,2,2,2,2
10	1,2,3,5,7,9	2,2,2,2,2,2
11	1,2,4,6,8,10	2,2,2,2,2,2
12	1,3,5,7,9,11	2,2,2,2,2,2
13	2,4,5,8,10,12	2,2,2,2,2,2
14	3,5,7,9,11,13	2,2,2,2,2,2
15	4,6,8,10,12,14	2,2,2,2,2,2

Table 1. Pooling layer configuration for 3DCNN variants

terpolation, then fed into the network in groups of T consecutive frames. The network outputs a T-length pulse waveform. In this work, the parameter T was selected to be 136 frames as this value typically captures at least one full heart cycle, yet fits on GPUs up to the largest depth network evaluated, and 3DCNN has been shown to exhibit relatively little performance variability for values of T between 32 and 256 frames [25].

We generated variations of 3DCNN for network depths of 2 to 15 Conv3d layers. In particular, as with the default 3DCNN, the first Conv3d layer has a (1,5,5) kernel size and 32 output channels, all intermediate Conv3d layers are (5,3,3) with 64 output channels, and the last is (1,1,1) with 1 output channel. All Conv3d layers except for the final layer are followed by a batch normalization layer and a ReLU. Odd numbered layers other than the first layer are further followed by a drop3d layer configured at p=0.5.

In order to accommodate varying numbers of layers, we organize pooling layers after the ReLU or drop3d (if applicable) of certain conv3d layers as outlined in Table 1. In each case, the final pooling layer is an average pooling layer, while all others are maximum pooling layers.

3.1.2 TS-CAN

The Temporal Shift Convolutional Attention Network (TS-CAN) [9] was developed to jointly probe spatial and temporal features in video data. In particular, input video is cropped around the face and downsampled to 36×36 pixels, then the pairwise differences between frames are calculated. Both the raw frames and the pairwise differences are fed into the network in 20-frame segments. TS-CAN can be trained to produce a single rPPG sequence, or multiple sequences for multi-task learning. In our experiments we focus on the single-task rPPG problem.

Depth	Pooling Indices
1	1
2	1,2
3	1,2,3
4	1,2,3,4
5	1,3,4,5
6	1,3,5,6
7	1,3
8	1,3
9	1,3
10	1

Table 2. Pooling layer configuration for TS-CAN variants

We generated TS-CAN models of varying depths by grouping a set of operations into a "meta-layer", which we then repeat to depths of 1 to 10 meta-layers. The grouping is as follows:

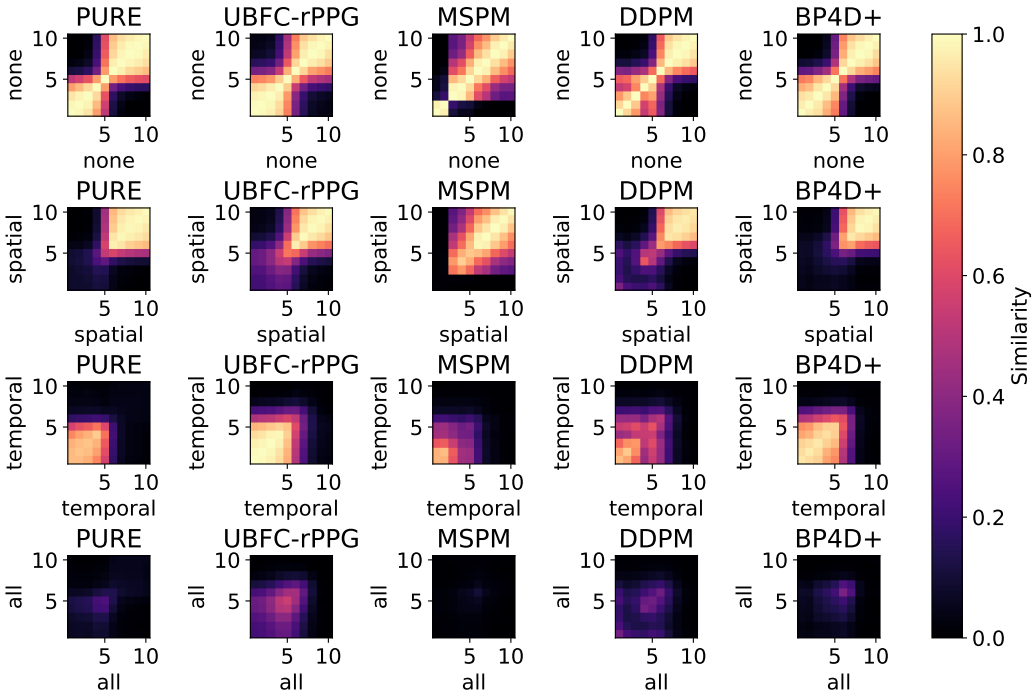
- Diff branch: TSM, Conv2d, tanh, TSM, Conv2d, tanh.
- Raw branch: Conv2d, tanh, Conv2d, tanh.
- Mixing branch: Conv2d (of raw branch output), sigmoid, attention mask, and results are multiplied by the diff branch output.
- Diff branch: average pooling (of mixing branch output), dropout.
- Raw branch: average pooling, dropout.

We configure layers in these groupings identically to the published TS-CAN model, an implementation of which is available in the rPPG-Toolbox [10].

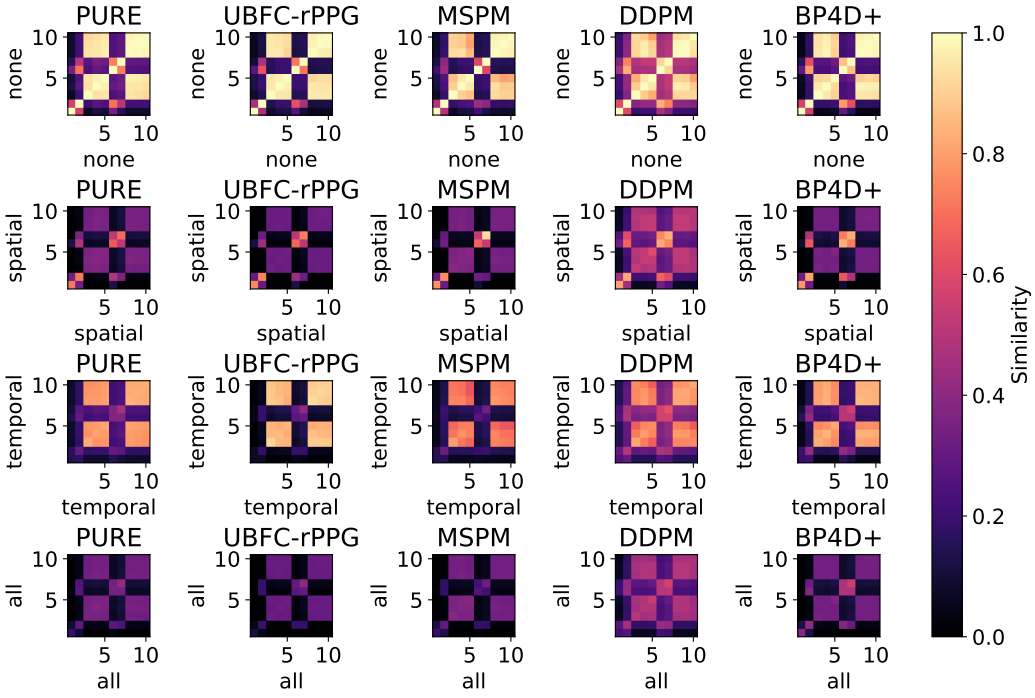
The TS-CAN architecture yields a decrease in spatial resolution at every meta-layer. As a result, in order to accommodate deeper variants we both increase the size of input video frames from the published $36x36$ pixels to $64x64$ pixels, and we constrain average pooling to select layers as given in Table 2.

3.2. CKA Analysis

We performed a Centered Kernel Alignment (CKA) analysis for each model depth to understand network pathology. After [5], we hypothesize that strong similarities between different layers of the same model indicate redundancies in the architecture such that the number of layers may be reduced without a large performance degradation. Furthermore, we hypothesize that highly similar layers between different architectures perform similar rPPG tasks, and that any layers without a corresponding similar layer in the other architecture may perform a task not handled by the other architecture. Examples of such groupings of similar and dissimilar layers are shown in Figure 1, and are expounded upon and analyzed in Section 4.1.



(a) 3DCNN



(b) TS-CAN

Figure 2. CKA comparison across augmentations, datasets, and architectures, demonstrating that blocks of layers exhibit similar behavior.

We provide an illustrative example of how CKA highlights model behavior. We trained each of the investigated architectures at their published depths (*i.e.*, 3DCNN-10 and TS-CAN-2). The training was performed over a variety of datasets that are described in Section 4, and we performed CKA analyses over them with different sets of transforma-

tions intended to showcase how portions of the network operate. In particular, this analysis comprises the following transformation sets:

- **none**: Perform no transformations.
- **spatial**: Perform spatial transformations: randomly flip, add illumination noise, and Gaussian blur.
- **temporal**: Vary the playback speed and modulate the change in playback speed.
- **all**: Combine both spatial and temporal transformations.

Figure 2 depicts CKA maps comparing the similarity of models. In particular, the similarity of activations of every layer in a model on the x axis is mapped against every layer of the model on the y axis. The numeric value on each axis corresponds to the layer index in the architecture as described in Section 3.1. Layer comparisons with a high degree of similarity result in a lighter color, whereas layer comparisons that are less similar result in a darker color.

Different portions of the network are affected differently by these transformations. In each column of Figure 2, a different dataset is explored, and in each row the model is compared to itself while undergoing one of four sets of transformations.

In Figure 2a we observe that, across all datasets, the 3DCNN network adopts a block structure with two main regions which vary in size depending on the training dataset, as most clearly presented in the comparisons with no transformations (the first row of Figure 2a). When the spatial transformations are applied, the similarity of the earlier region is reduced, while the latter region remains relatively unaffected. Similarly, the temporal transformations affect the latter region more heavily than the early region. When both transformation sets are applied the full model is affected. These results indicate that across datasets the 3DCNN model learns an internal structure in which its early layers process spatial features, while its latter layers process temporal features.

In Figure 2b we observe a different internal structure in TS-CAN than in 3DCNN. This is due to the dual-branch architecture of TS-CAN, in which the first two layers are part of the "Diff" branch dealing with temporal features, then the next three layers are part of the "Raw" branch dealing with spatial features, then the pattern repeats for the second meta-layer. This is observed in the spatial row of Figure 2b in which the similarity within the "Raw" blocks is reduced while the "Diff" blocks are relatively less affected. Similarly the temporal augmentations heavily affect the "Diff" blocks while affecting the "Raw" blocks less severely.

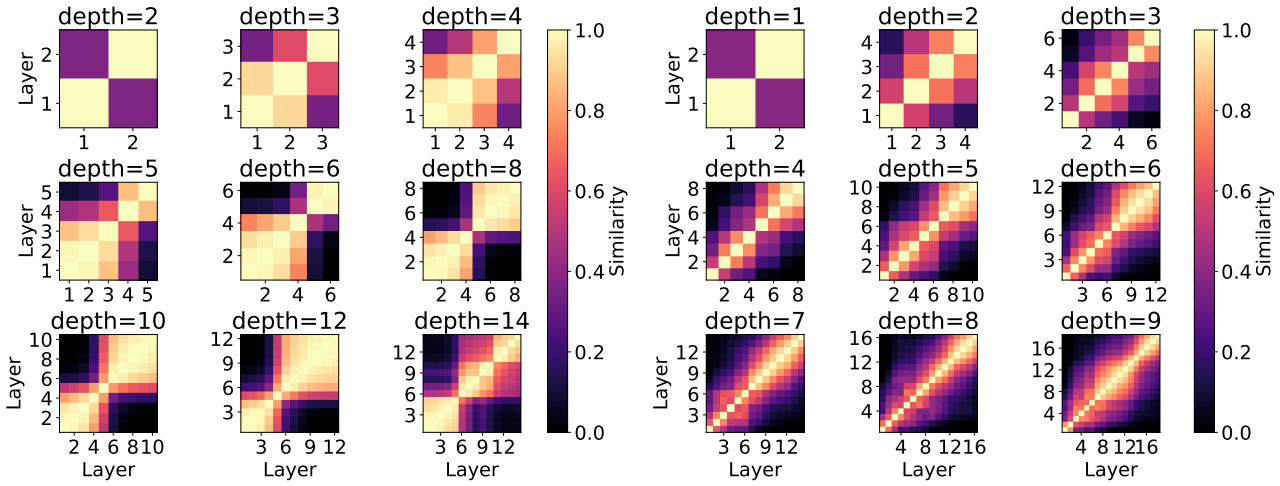
These analyses indicate that both 3DCNN and TS-CAN have internal structures in which some layers process temporal features while others process spatial features. In the case of 3DCNN, the spatial processing layers occur early in the network, temporal layers occur later in the network, and the precise divide appears to be learned from the data. In the case of TS-CAN, the assignment of tasks is a facet of the dual-branch architecture. Furthermore, these analyses demonstrate how CKA reveals behavioral information regarding how models process data.

4. Results

We trained models based on 3DCNN and TS-CAN at different depths according to the architecture definitions given in Section 3.1. These models were then analyzed with CKA and also used in pulse estimation experiments to determine if architectural under/overprovisioning is reflected in empirical results. For a robust analysis, we investigated the following datasets:

- **PURE** [18] is a small dataset of 10 subjects with six one-minute videos, each with constrained head motions.
- **UBFC-rPPG** [1] is a 43 subject dataset with an average of 1.6 minutes of video for each subject. In each video, the subject played a time-sensitive mathematical game intended to elicit an elevated heart rate.
- **DDPM** [21] is a large 93 subject dataset with between 8 and 11 minutes of video for each subject. In each video, subjects engaged in a mock-interview in which they attempted to deceive the interviewer on selected questions.
- **MSPM** [13] is a large 103 subject dataset with an average of 14 minutes of video for each subject. In each video, subjects engaged in a sequence of activities including a breathing exercise, playing a racing game, and watching videos. The dataset includes an adversarial attack in which pulsating colored light is projected onto the subjects — we omit this portion of the dataset in our analyses (the adversarial attack succeeds in obliterating the pulse waveform in its interval).
- **BP4D+** [27] is a large 140 subject dataset with an average of 9 minutes of video for each subject. This dataset is a spontaneous emotion corpus in which each subject experiences 10 different activities designed to elicit different emotions (*e.g.* embarrassment due to improvising a silly song, or startle/surprise due to experiencing a sudden burst of sound). It is collected with a continuous blood pressure monitoring for its blood volume pulse ground truth, whereas the other datasets use a pulse oximeter.

We trained most models for 40 epochs. 3DCNN variants with depths of 14-15 being trained on DDPM and MSPM required 80 epochs for the loss to reach a plateau. We trained using the same set of augmentations as used in [22], in which video clips are scaled temporally to capture a broad band of heart rate frequencies. This augmentation has been shown to promote generalization rather than memorization — an important feature for meaningful network pathology analysis [12]. We use a negative Pearson loss function, k-fold cross validation with k=5, the Adam optimizer with a learning rate of 0.0001, and validation loss for model selection.



(a) 3DCNN (b) TS-CAN

Figure 3. CKA self-similarity comparison for 3DCNN (3a) and TS-CAN (3b) based architectures on the PURE dataset.

4.1. CKA-based Analysis

We investigate the representations of the data by the networks using CKA as described in Section 3.2. Figure 3a shows a CKA self-comparison of 3DCNN-based models trained on the PURE dataset. This and all other CKA plots in this paper depict similarities that are averaged across the 5 folds, and we did not observe any significant differences between folds that this averaging would mask. We observe that the deeper 3DCNN variants tend to have two or three blocks of highly similar layers. This suggests that there are a limited number of distinct sections of the network performing discrete tasks, each of which merely gains new layers in a piecemeal fashion as additional layers are added.

Figure 4a confirms that these distinct model sections perform the same function even across architectures of differing depths. We begin the cross-architecture comparison with the 10-layer model because that is the depth of the published PhysNet-3DCNN architecture. In this 10-layer model, we observe three distinct regions: layers 1-4, layer 5, and layers 6-10. Interestingly, it does not appear that these regions are present at every depth, but rather that they are added only in sufficiently deep models: while the region composed of layers 1-4 has strongly similar counterparts in all compared models, the region composed of layers 6-10 does not have highly similar regions in models of 4 layers or shallower, while layer 5 has only weakly similar counterparts in shallower models. This may indicate that models of depths 2-4 are not sufficiently parameterized to gain the functionality of the latter parts of the 10-layer model.

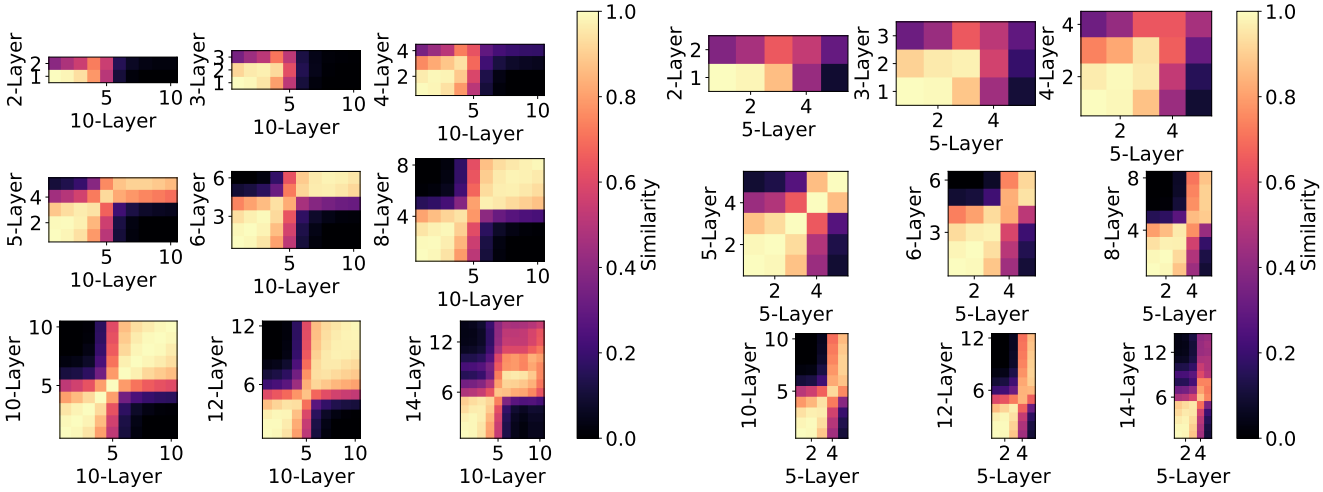
Indeed, when comparing the 5-layer model to the other architectures in Figure 4b, we observe that the model region in layer 4 has only weak counterparts in less-deep architectures, while layer 5 does not appear to be represented at all. In contrast, when comparing the 5-layer model to deeper architectures, it appears to strongly capture the latter regions of these architectures until reaching the 14-layer model. Due to these observations, we suspect that the 5-layer model sufficiently captures the representations present in deeper models, yet without as many redundancies.

We continue our analysis by investigating the TS-CAN architecture in Figure 3b. Because we are focused exclusively on the temporal rPPG problem, whereas TS-CAN was developed to handle multi-task learning of both appearance and temporal features, we constrain our analysis to the temporal branch, which contains two Conv2d layers for every replicated TS-CAN layer in depth. Unlike the 3DCNN architecture, TS-CAN exhibits a strong CKA diagonal with only a subtle block structure visible in Figure 3b. This may indicate that deeper variants of TS-CAN will learn more detailed representations of the data.

We continue our analysis comparing the 2-metalayer TS-CAN architecture (i.e., the published architecture) to other depths in Figure 5. We observe that the four Conv2d layers present exhibit the strongest similarity to the first four Conv2d layers in each deeper architecture, with low similarity to the deepest layers in the deeper achitectures. This corroborates with the self-similarity observation, that deeper TS-CAN variants appear to learn representations that are not learned by the 2-metalayer architecture.

4.2. Empirical Error-based Analysis

We perform an empirical study to test our findings from Section 4.1. Figures 6a-6e show the Mean Absolute Error (MAE) for 3DCNN-based networks of depths 2-15 on the investigated datasets. We observe that shallower models tend to exhibit reduced accuracy and greater variation



(a) 10-to-all (b) 5-to-all

Figure 4. CKA 10-to-all (4a) and 5-to-all (4b) cross-similarity comparison for 3DCNN-based architectures on the PURE dataset.

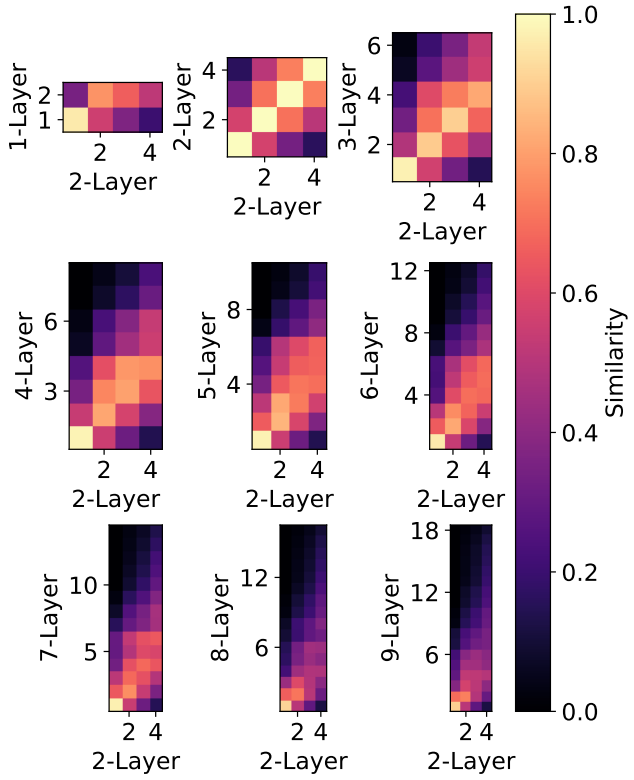
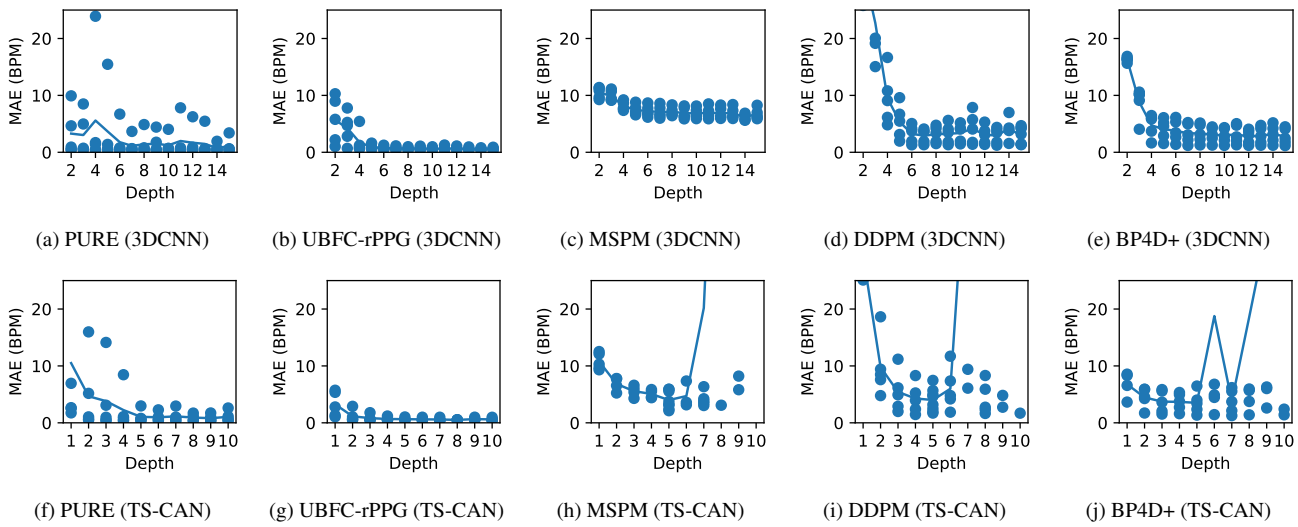


Figure 5. CKA 2-to-all cross-similarity comparison for TS-CAN based architectures on the PURE dataset.

in accuracy than deeper models, corroborating the CKA diagnostic that these shallower models do not have the same level of functionality as the deeper variants. Furthermore, no significant gains appear to be made for models deeper than four layers for BP4D+ (Figure 6e), five layers for UBFC-rPPG (Figure 6b), or six layers for PURE (Figure 6a) and DDPM (Figure 6d), corroborating the findings from the CKA analysis. There does appear to be an insignificant improvement in MSPM at 14 layers (Figure 6c), which could be due to the third block of layers suggested by CKA that emerges in the latter layers of 14-layer models.

We additionally test our CKA findings on TS-CAN architectures with rPPG experiments, with results documented in Figures 6f-6j. Our CKA findings suggested that a TS-CAN architecture of only 2 metalayers may be underprovisioned, with deeper variants learning representations that are not present in shallower models, resulting in generally lower empirical errors up to a depth of about 5 metalayers. Beyond this depth, we found that models had difficulty converging on MSPM, DDPM, and BP4D+. These datasets are larger than PURE and UBFC-rPPG by over an order of magnitude (1480, 776, and 1285 minutes for MSPM, DDPM, and BP4D+ respectively, and 60 and 70 minutes for PURE and UBFC respectively). They are also more complex, exhibiting conversation, unconstrained head movement, and activities designed to produce large fluctuations in heart rate. Meanwhile, the dual-branched TS-CAN architecture was designed to contain only two metalayers, yet we have abused its design by replicating its highly engineered structure to several times its original depth. Though we attempted extending training from 40 to 80 epochs as was done with deeper architectures based on 3DCNN, this did not result in model convergence. We believe it likely that reliably training deeper versions of TS-CAN on these more complicated datasets may require adding skip connections, adjusting dropout probabilities, learning rate scheduling, or other techniques for model convergence of deep net-



(a) PURE (3DCNN) (b) UBFC-rPPG (3DCNN) (c) MSPM (3DCNN) (d) DDPM (3DCNN) (e) BP4D+ (3DCNN)

(f) PURE (TS-CAN) (g) UBFC-rPPG (TS-CAN) (h) MSPM (TS-CAN) (i) DDPM (TS-CAN) (j) BP4D+ (TS-CAN)

Figure 6. Empirical Results for architectures based on 3DCNN (Figures 6a-6e) and TS-CAN (Figures 6f-6j). For visualization purposes the y axis was constrained to errors under 25 BPM, which resulted in truncated results for Depths 2 and 3 for 6d and depth 1 in 6f and 6i. Depths 7-10 in 6h and 6i, and Depths 6 and 8-10 in 6j signal training divergence at those depths (see text for comments).

works.

The results for PURE in Figures 6a and 6f exhibit severe errors in the split containing a subject with a low heart rate and strong dichrotic notch. This contributed the highest errors for this dataset for both networks of every depth other than 3DCNN at depths 2 and 3, for which it was the 2nd highest.

5. Conclusions

We performed an in-depth CKA analysis on two rPPG model architectures (3DCNN and TS-CAN) over a range of architecture depths. We showed that CKA is useful for understanding model representations, both in terms of how layers are similar or unique within a model as well as across architectures, thereby informing architecture selection.

Our results, both utilizing CKA and by empirical errors across five rPPG datasets, suggest that the investigated architectures may be refined in terms of depth: The published 3DCNN depth of 10 layers appears to be deeper than necessary, with only 5 layers maintaining highly similar CKA model representations and 6 layers achieving comparable empirical performance. Similarly, the published TS-CAN depth of 2 metalayers appears to be shallower than optimal, with deeper models learning new representations not present in the published model, and empirical results showing an improved performance up to 5 metalayers.

We believe that future work utilizing CKA for architecture insights should extend in at least two dimensions. First, CKA can be used to inform more than just model depth. For example, we observe that the 3DCNN architecture exhibits a block structure at most depths, where the primary functions of the network are constrained to only a few large blocks. This is indicative that architectural adjustments other than network depth may prove fruitful in reducing model complexity without reducing performance, e.g., adjusting pooling layers, kernel sizes, or channels. Secondly, while our experiments focused on comparisons across architectures while using different datasets to validate our results, we believe that valuable insight could be gained by comparing models trained on different datasets using CKA with regards to dataset similarity under the investigated architecture and training regime.

References

[1] Serge Bobbia, Richard Macwan, Yannick Benezeth, Alamin Mansouri, and Julien Dubois. Unsupervised skin tissue segmentation for remote photoplethysmography. *Pattern Recognition Letters*, 124:82–90, 2019. 5

[2] Weixuan Chen and Daniel McDuff. DeepPhys: Video-based physiological measurement using convolutional attention networks. In *European Conference on Computer Vision (ECCV)*, pages 356–373, 2018. 2

[3] Tianyu Cui, Yogesh Kumar, Pekka Marttinen, and Samuel Kaski. Deconfounded representation similarity for comparison of neural networks. *Advances in Neural Information Processing Systems*, 35:19138–19151, 2022. 2

[4] G. de Haan and V. Jeanne. Robust pulse rate from chrominance-based rppg. *IEEE Trans. on Biom. Eng.*, 60 (10):2878–2886, 2013. 2

[5] Simon Kornblith, Mohammad Norouzi, Honglak Lee, and Geoffrey Hinton. Similarity of neural network representations revisited. In *International conference on machine learning*, pages 3519–3529. PMLR, 2019. 1, 2, 3

[6] Nikolaus Kriegeskorte, Marieke Mur, and Peter A Bandettini. Representational similarity analysis-connecting the branches of systems neuroscience. *Frontiers in systems neuroscience*, 2:249, 2008. 2

[7] Aarre Laakso and Garrison Cottrell. Content and cluster analysis: assessing representational similarity in neural systems. *Philosophical psychology*, 13(1):47–76, 2000. 2

[8] Eugene Lee, Evan Chen, and Chen-Yi Lee. Meta-rppg: Remote heart rate estimation using a transductive meta-learner. In *European Conference on Computer Vision (ECCV)*, 2020. 2

[9] Xin Liu, Josh Fromm, Shwetak Patel, and Daniel McDuff. Multi-task temporal shift attention networks for on-device contactless vitals measurement. *Advances in Neural Information Processing Systems*, 33:19400–19411, 2020. 1, 2, 3

[10] Xin Liu, Girish Narayanswamy, Akshay Paruchuri, Xiaoyu Zhang, Jiankai Tang, Yuzhe Zhang, Yuntao Wang, Soumyadip Sengupta, Shwetak Patel, and Daniel McDuff. rppg-toolbox: Deep remote ppg toolbox. *arXiv preprint arXiv:2210.00716*, 2022. 2, 3

[11] Hao Lu, Hu Han, and S Kevin Zhou. Dual-gan: Joint bvp and noise modeling for remote physiological measurement. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 12404–12413, 2021. 2

[12] Ari Morcos, Maithra Raghu, and Samy Bengio. Insights on representational similarity in neural networks with canonical correlation. *Advances in neural information processing systems*, 31, 2018. 2, 5

[13] Lu Niu, Jeremy Speth, Nathan Vance, Benjamin Sporrer, Adam Czajka, and Patrick Flynn. Full-body cardiovascular sensing with remote photoplethysmography. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5993–6003, 2023. 5

[14] Xuesong Niu, Shiguang Shan, Hu Han, and Xilin Chen. RhythmNet: End-to-end heart rate estimation from face via spatial-temporal representation. *IEEE Transactions on Image Processing*, 29:2409–2423, 2020. 2

[15] Ming-Zher Poh, Daniel J McDuff, and Rosalind W Picard. Non-contact, automated cardiac pulse measurements using video imaging and blind source separation. *Optics express*, 18(10):10762–10774, 2010. 2

[16] Maithra Raghu, Justin Gilmer, Jason Yosinski, and Jascha Sohl-Dickstein. Svcca: Singular vector canonical correlation analysis for deep learning dynamics and interpretability. *Advances in neural information processing systems*, 30, 2017. 2

[17] Jeremy Speth, Nathan Vance, Patrick Flynn, Kevin Bowyer, and Adam Czajka. Unifying frame rate and temporal dilations for improved remote pulse detection. *Computer Vision and Image Understanding*, 210:103246, 2021. 1, 2

[18] Ronny Stricker, Steffen Muller, and Horst Michael Gross. Non-contact video-based pulse rate measurement on a mobile service robot. *IEEE International Symposium on Robot and Human Interactive Communication*, pages 1056–1062, 2014. 5

[19] Chihiro Takano and Yuji Ohta. Heart rate measurement based on a time-lapse image. *Medical engineering & physics*, 29(8):853–857, 2007. 2

[20] Yun-Yun Tsou, Yi-An Lee, Chiou-Ting Hsu, and Shang-Hung Chang. Siamese-rppg network: Remote photoplethysmography signal estimation from face videos. In *Proceedings of the 35th annual ACM symposium on applied computing*, pages 2066–2073, 2020. 2

[21] Nathan Vance, Jeremy Speth, Siamul Khan, Adam Czajka, Kevin W. Bowyer, Diane Wright, and Patrick Flynn. Deception detection and remote physiological monitoring: A dataset and baseline experimental results. *IEEE Transactions on Biometrics, Behavior, and Identity Science (TBIOM)*, pages 1–1, 2022. 5

[22] Nathan Vance, Jeremy Speth, Benjamin Sporrer, and Patrick Flynn. Promoting generalization in cross-dataset remote photoplethysmography. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5984–5992, 2023. 5

[23] Wim Verkruysse, Lars O Svaasand, and J Stuart Nelson. Remote plethysmographic imaging using ambient light. *Optics express*, 16(26):21434–21445, 2008. 2

[24] W. Wang, A. C. den Brinker, S. Stuijk, and G. de Haan. Algorithmic principles of remote ppg. *IEEE Trans. on Biom. Eng.*, 64(7):1479–1491, 2017. 2

[25] Zitong Yu, Xiaobai Li, and Guoying Zhao. Remote photoplethysmograph signal measurement from facial videos using spatio-temporal networks. *arXiv preprint arXiv:1905.02419*, 2019. 1, 2, 3

[26] Zitong Yu, Yuming Shen, Jingang Shi, Hengshuang Zhao, Philip H.S. Torr, and Guoying Zhao. Physformer: Facial video-based physiological measurement with temporal difference transformer. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4186–4196, 2022. 2

[27] Zheng Zhang, Jeff M Girard, Yue Wu, Xing Zhang, Peng Liu, Umur Ciftci, Shaun Canavan, Michael Reale, Andy Horowitz, Huiyuan Yang, et al. Multimodal spontaneous emotion corpus for human behavior analysis. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3438–3446, 2016. 5

[28] Yu Zhao, Bochao Zou, Fan Yang, Lin Lu, Abdelkader Nasreddine Belkacem, and Chao Chen. Video-based physiological measurement using 3d central difference convolution attention network. In *2021 IEEE International Joint Conference on Biometrics (IJCB)*, pages 1–6, 2021. 2