Multi-view Convolutional Neural Networks for 3D Shape Recognition

by

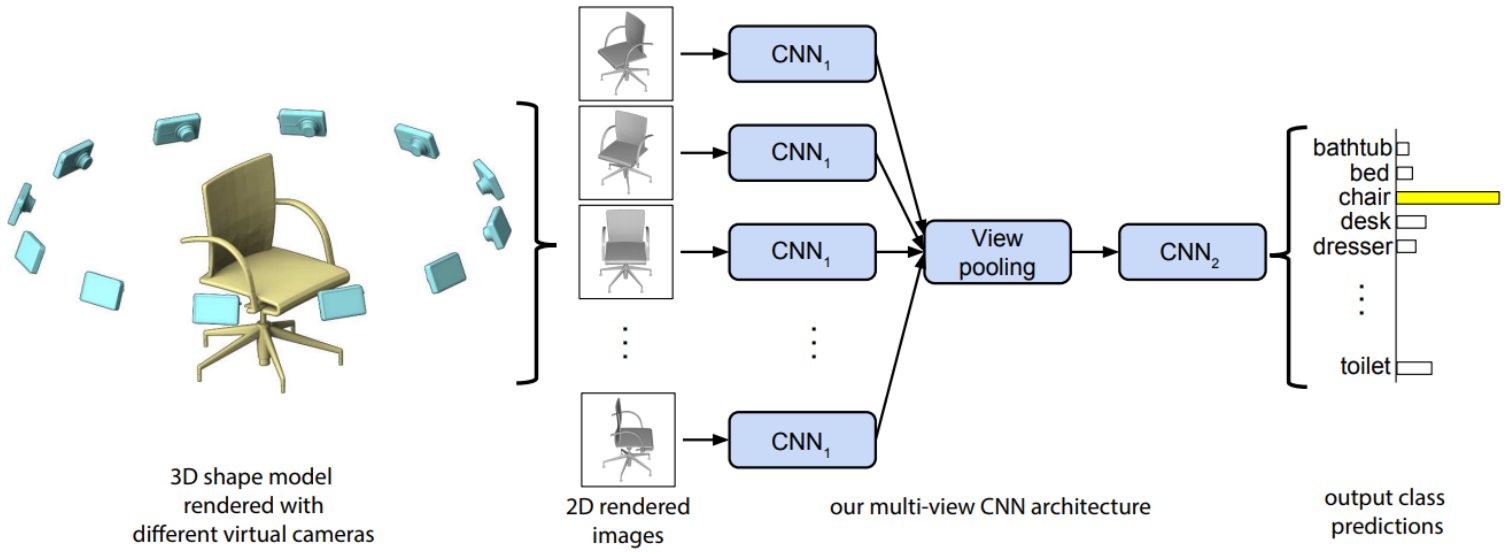
Hang Su Subhransu | Maji Evangelos Kalogerakis | Erik Learned-Miller
University of Massachusetts, Amherst

Presented by

Johannes Andersen

Overview

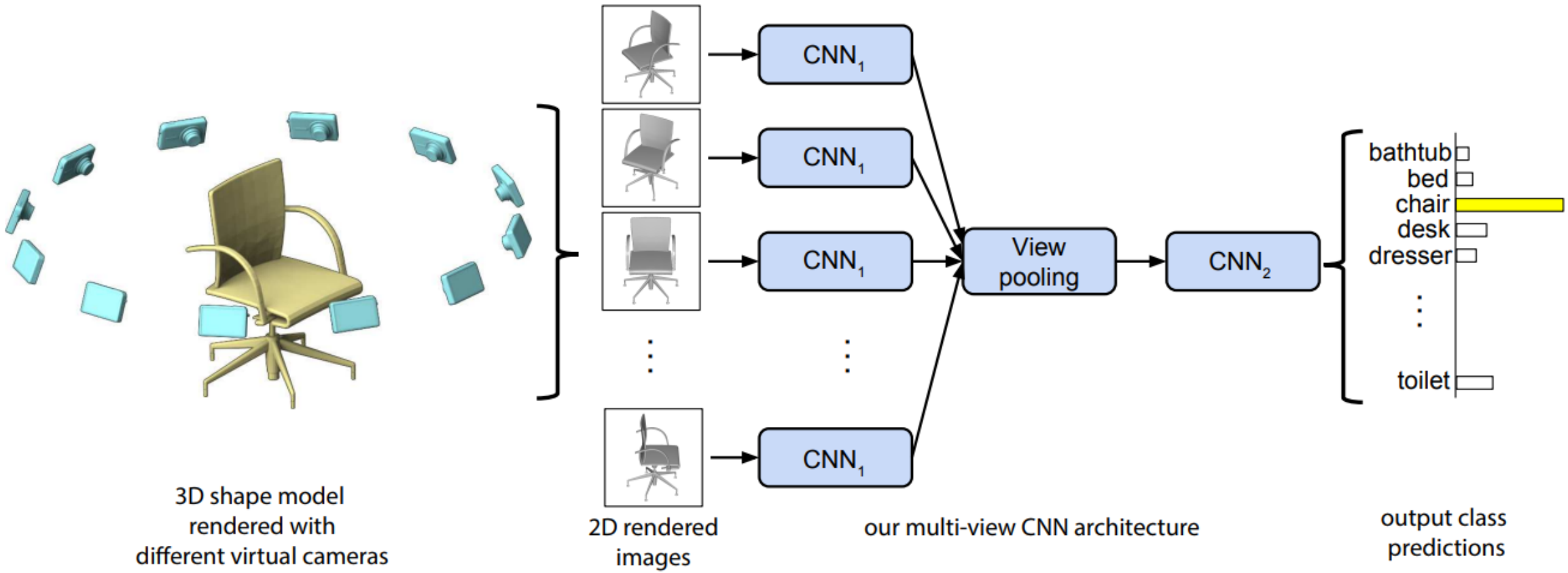
- Problem
- Solution
- Usecases
- Conclusion
- TLDR



Problem

- Representing and recognizing 3D models in CNN's:
  - High dimensionality (overfitting)
  - Lower resolution (compared to 2D)
  - Fewer viable image descriptors available in 3D
  - Fewer and smaller pretrained datasets than in 2D
  - Even rendering 3D shape at 1 view was better than 3D recognition

Solution


CNN1
CNN1
CNN1
CNN1
View pooling
CNN2
bathtub
bed
chair
desk
dresser
toilet
3D shape model rendered with different virtual cameras
2D rendered images
our multi-view CNN architecture
output class predictions

Solution

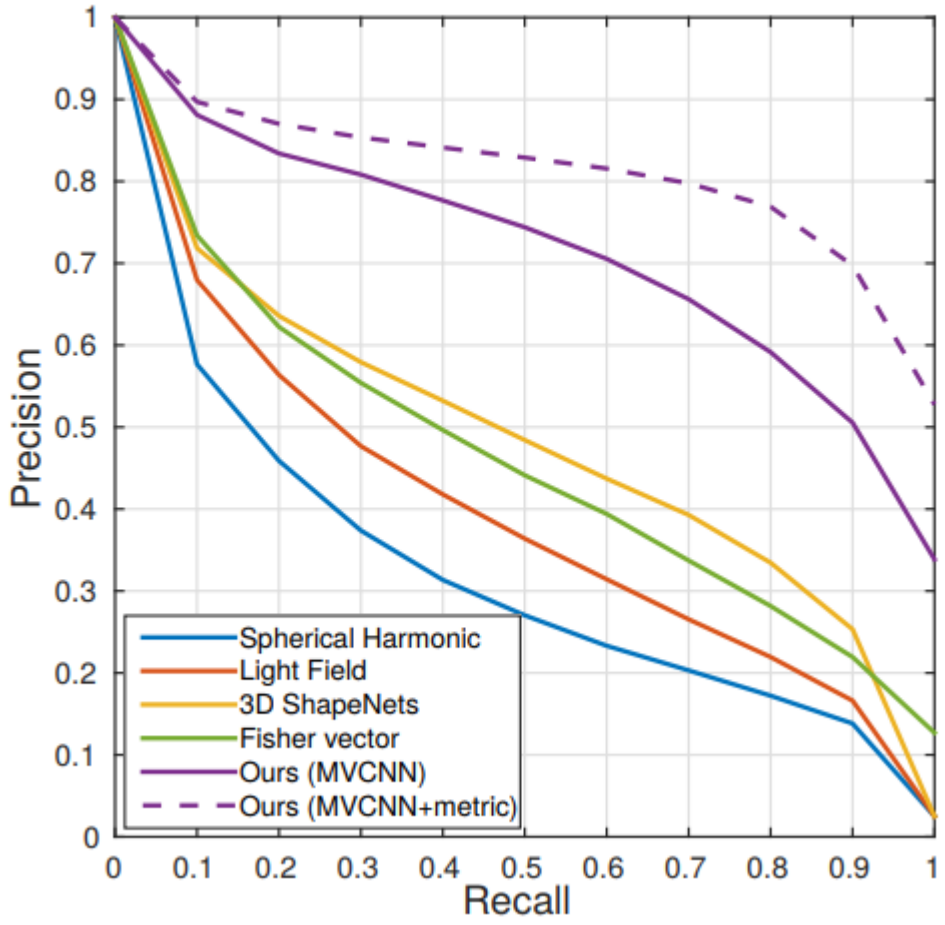
- Step one: Acquire 2D images of model from several views
- Step two: Create descriptor by CNN based on views (MVCNN)
- Step three:
  - Recognize 3D shapes using another CNN
  - Recognize sketches
  - 3D shape retrieval

Usecase: Recognizing 3D shapes

- Single-view: 78.8% acc.
- 12-view: 89.5%
- 3D descriptor: 77.3% acc.

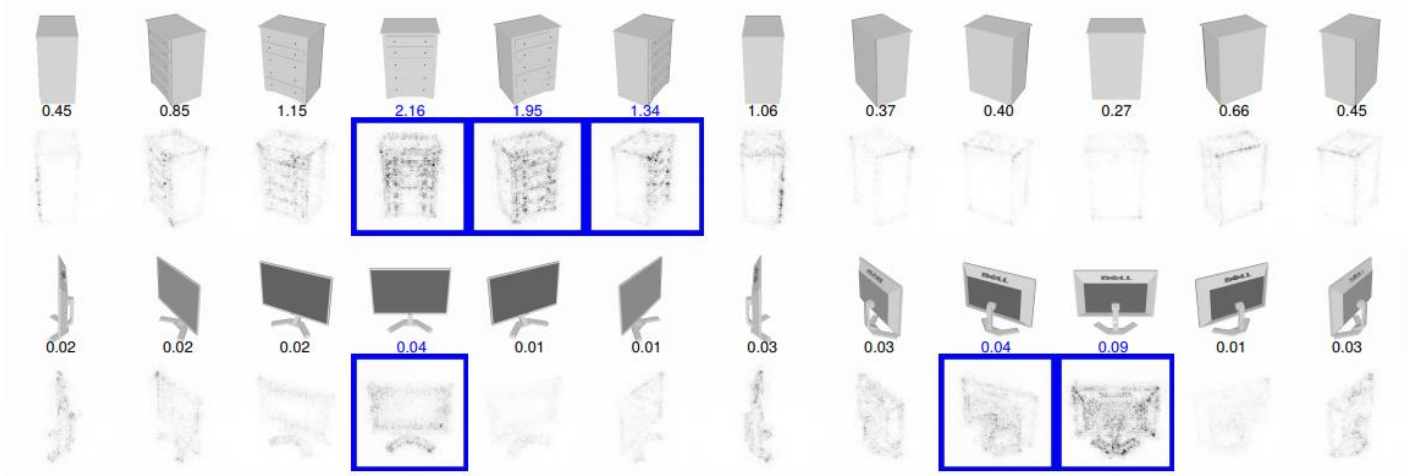
Method	Training Config.			Test Config.	Classification (Accuracy)	Retrieval (mAP)
	Pre-train	Fine-tune	#Views	#Views		
(1) SPH [16]	-	-	-	-	68.2%	33.3%
(2) LFD [5]	-	-	-	-	75.5%	40.9%
(3) 3D ShapeNets [37]	ModelNet40	ModelNet40	-	-	77.3%	49.2%
(4) FV	-	ModelNet40	12	1	78.8%	37.5%
(5) FV, 12×	-	ModelNet40	12	12	84.8%	43.9%
(6) CNN	ImageNet1K	-	-	1	83.0%	44.1%
(7) CNN, f.t.	ImageNet1K	ModelNet40	12	1	85.1%	61.7%
(8) CNN, 12×	ImageNet1K	-	-	12	87.5%	49.6%
(9) CNN, f.t.,12×	ImageNet1K	ModelNet40	12	12	88.6%	62.8%
(10) MVCNN, 12×	ImageNet1K	-	-	12	88.1%	49.4%
(11) MVCNN, f.t., 12×	ImageNet1K	ModelNet40	12	12	89.9%	70.1%
(12) MVCNN, f.t.+metric, 12×	ImageNet1K	ModelNet40	12	12	89.5%	**80.2%**
(13) MVCNN, 80×	ImageNet1K	-	80	80	84.3%	36.8%
(14) MVCNN, f.t., 80×	ImageNet1K	ModelNet40	80	80	**90.1%**	70.4%
(15) MVCNN, f.t.+metric, 80×	ImageNet1K	ModelNet40	80	80	**90.1%**	79.5%

* f.t.=fine-tuning, metric=low-rank Mahalanobis metric learning



Usecase : Sketch recognition (“jittering”)

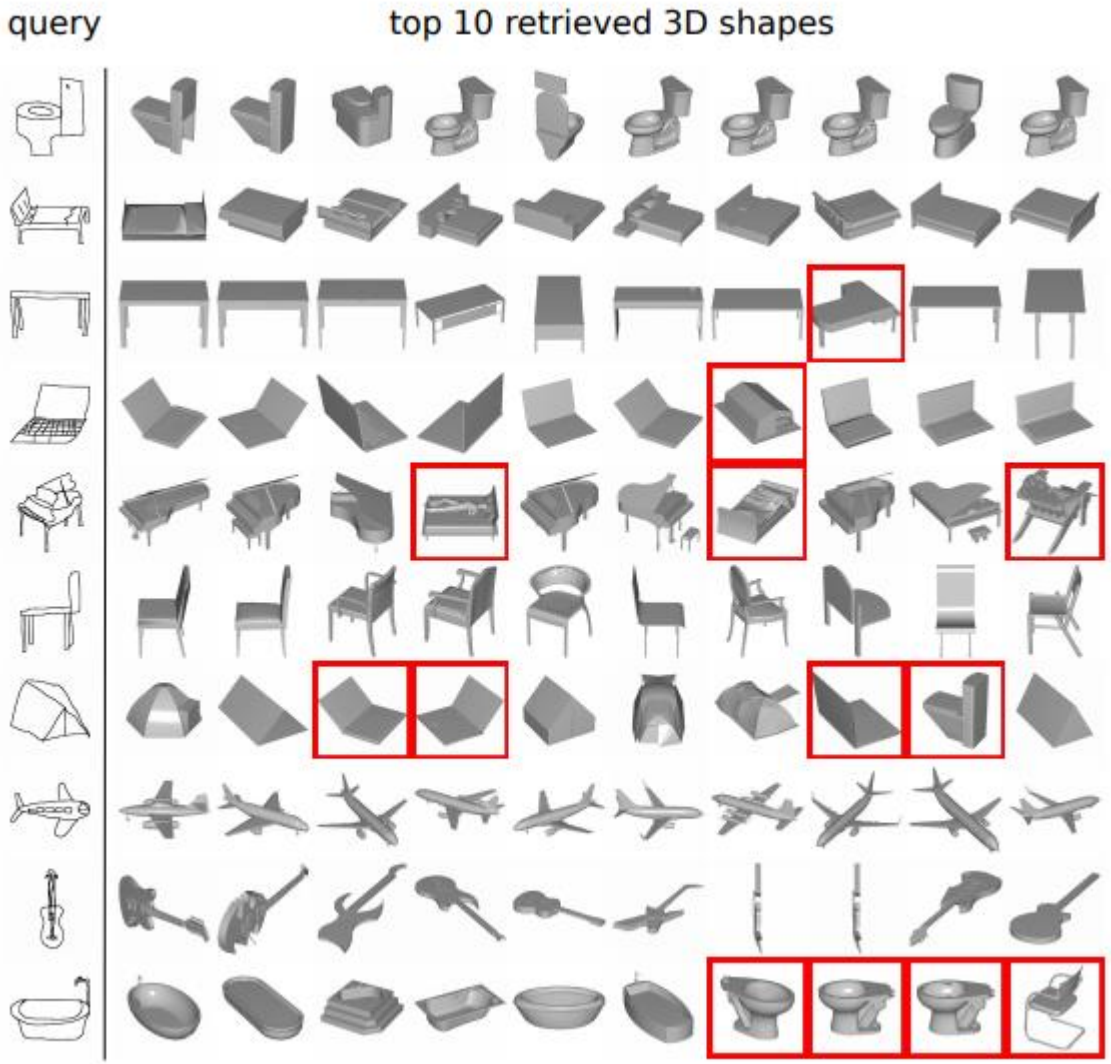
- VGG-M: 77.3% acc.
- VGG-VD (deeper): 86.0% acc.
- MVCNN VD: 87.2% acc.



Method	Aug.	Accuracy
(1) FV [30]	-	79.0%
(2) CNN M	-	77.3%
(3) CNN M, fine-tuned	-	84.0%
(4) CNN M, fine-tuned	6×	85.5%
(5) MVCNN M, fine-tuned	6×	86.3%
(6) CNN VD	-	69.3%
(7) CNN VD, fine-tuned	-	86.3%
(8) CNN VD, fine-tuned	6×	86.0%
(9) MVCNN VD, fine-tuned	6×	**87.2%**
(10) Human performance	n/a	93.0%

Usecase : 3D shape retrieval

- Red boxes are wrong

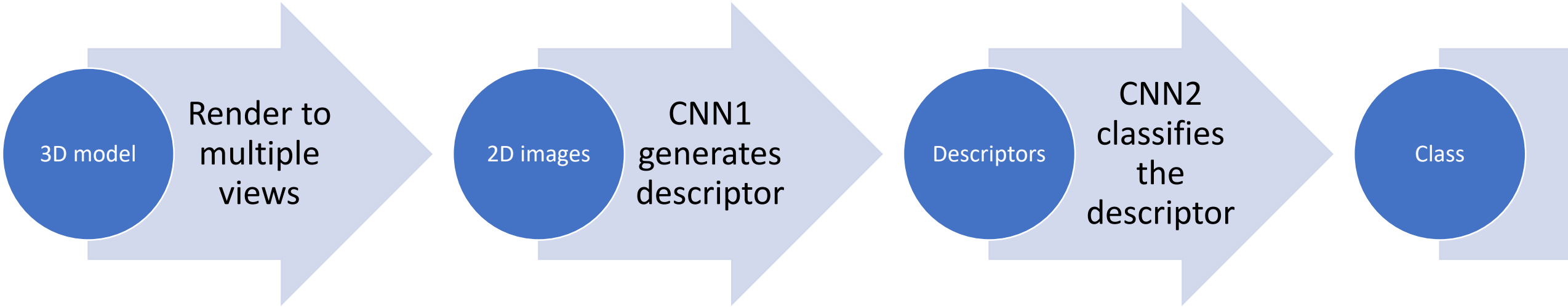


Conclusion

- 2D input to modern architectures outperform 3D representations
- Better accuracy through aggregated descriptors
- Can use these descriptors to find 3D models of 2D images

- Which views are most informative?
- How many views are needed?
- Can it be used in real-time with video?
- Can it select useful views on the fly?

TLDR


3D model
Render to multiple views
2D images
CNN1 generates descriptor
Descriptors
CNN2 classifies the descriptor
Class

Sources

- Paper: https://www.cv-foundation.org/openaccess/content_iccv_2015/papers/Su_Multi-View_Convolutional_Neural_ICCV_2015_paper.pdf
- Original implementation: https://github.com/suhangpro/mvcnn
- TensorFlow implementation: https://github.com/WeiTang114/MVCNN-TensorFlow