Hierarchical Feature Extraction for Efficient Design of Microfluidic Flow Patterns

Kin Gwn Lore, Daniel Stoecklein, Michael Davies, Baskar Ganapathysubramanian, Soumik Sarkar | Department of Mechanical Engineering, Iowa State University

Neural Information Processing Systems Foundation

NIPS 2015 Workshop
Feature Extraction:
Modern Questions and Challenges

Background

- Deep neural net architectures are used in:

Object Recognition	Speech Recognition
Scene Understanding	Biological & medical applications

- We apply **deep learning (DL) in design engineering** specifically in **microfluidic device or lab-on-a-chip design**.
- **Controlling shape** and **location** of fluid stream by using a sequence of pillars of various diameters and position in fluid channel (Fig 1, 2) enables creation of structured materials, preparing biological samples, and engineering heat and mass transport.
- Sculpting user-defined flow shapes for practical applications are **laborious** and **time-consuming** (require trial and error design iterations).

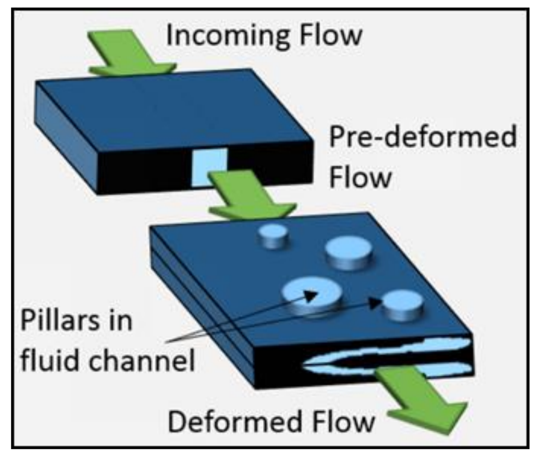


Fig 1: The forward problem: Sculpting fluid shapes with pillar sequences in fluid channel

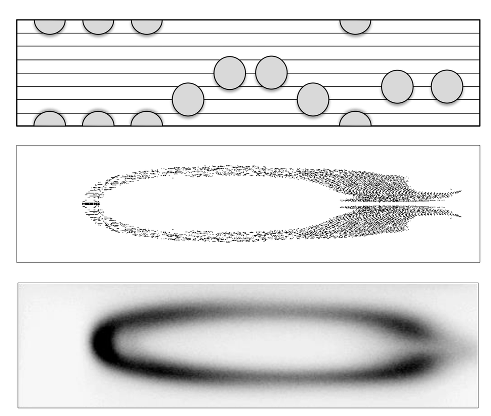


Fig 2: Pillar programs (top row), simulations (middle row) and experimental validation (bottom row).

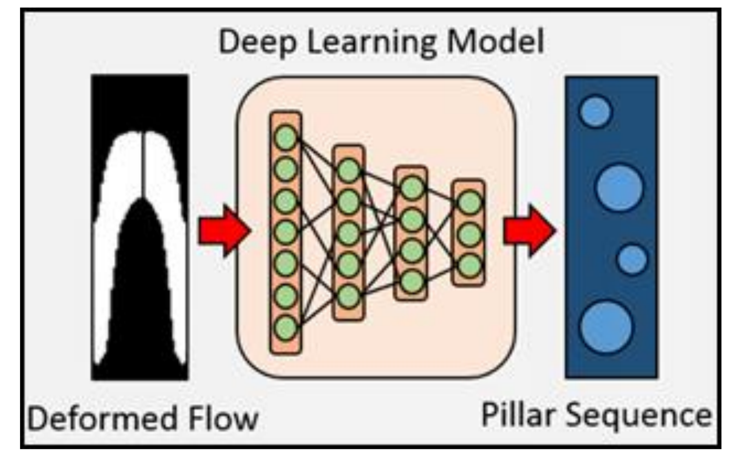


Fig 3: Solving the inverse problem with deep learning methods

- This framework **automatically determine a sequence of pillars** that yields this shape; using DL to solve this problem is perhaps one of the first applications in fluid mechanics.
- Standard techniques to solve the inverse design problem is successful but **time consuming**, hence not suited for real-time applications.

Contributions

- Hierarchical feature extraction provides a **scalable design tool** by learning semantic representations from a **relatively small** number of flow pattern **examples**
- Generating training data by **sampling the input space quasi-randomly** improves training performance
- Visualizations of intermediate layers/filters provide insights to the **diversity of data** in the input space
- Deep learning models are **much faster** than standard techniques and the current state-of-the-art in mapping user-defined flow shapes to corresponding sequence of pillars (Fig 3)

Problem Statement

The problem is solved by the following approach:

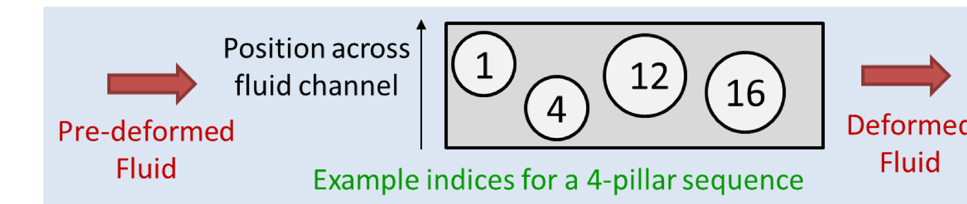


Fig 4: Discretization into indices

- **Discretizing pillar configuration as integers (Fig 4):** Combination of position and diameter of each pillar is assigned an index (or class). 4 diameters and 8 positions result in 32 classes.
- **Solving the problem by classification:** The input is the desired flow shape (that can be drawn freehand). A convolutional neural network (CNN) extract features from input images and predicts the index of each pillars in the sequence which produces the desired flow shape (Fig 5).

Challenges:

- **Curse of dimensionality:** Number of possible sequences increases exponentially as length of the sequence increases (Fig 7). More training data and intelligent data generation is required!

Solution:

- **On-the-fly data generation:** Provides sufficiently large volume of training data
- **Sobol-sequences for training data generation:** Instead of randomly sampling from the design space, using quasi-random Sobol sequences ensure more even coverage of the high-dimensional space.

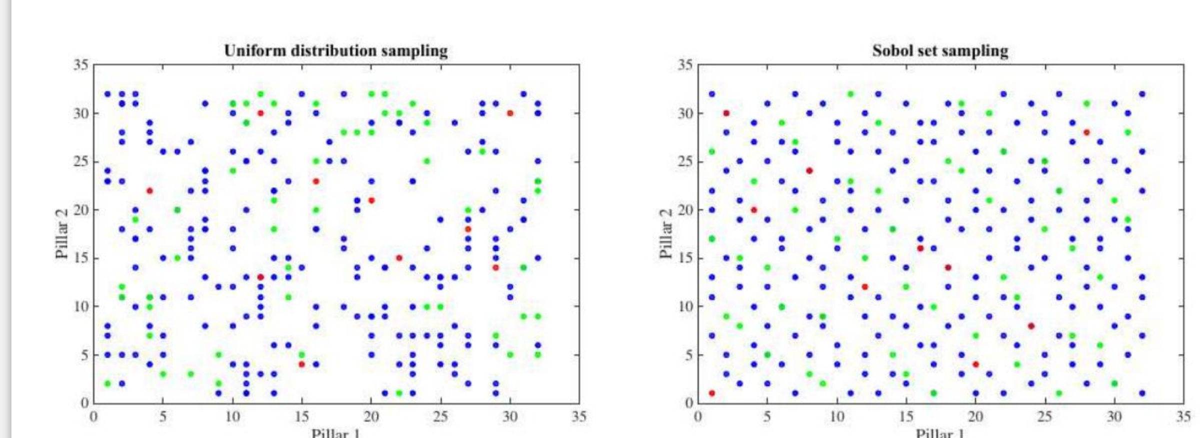


Fig 6 (top): Sobol Sampling

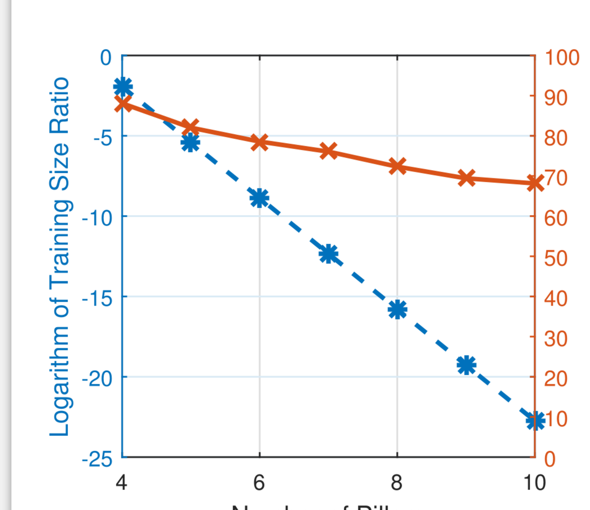


Fig 7 (left): The 10-pillar sequence has 32^{10} ($\sim 10^{15}$) possible configurations but achieves an acceptable ~70% prediction accuracy following the use of Sobol sequences.

Results & Discussions

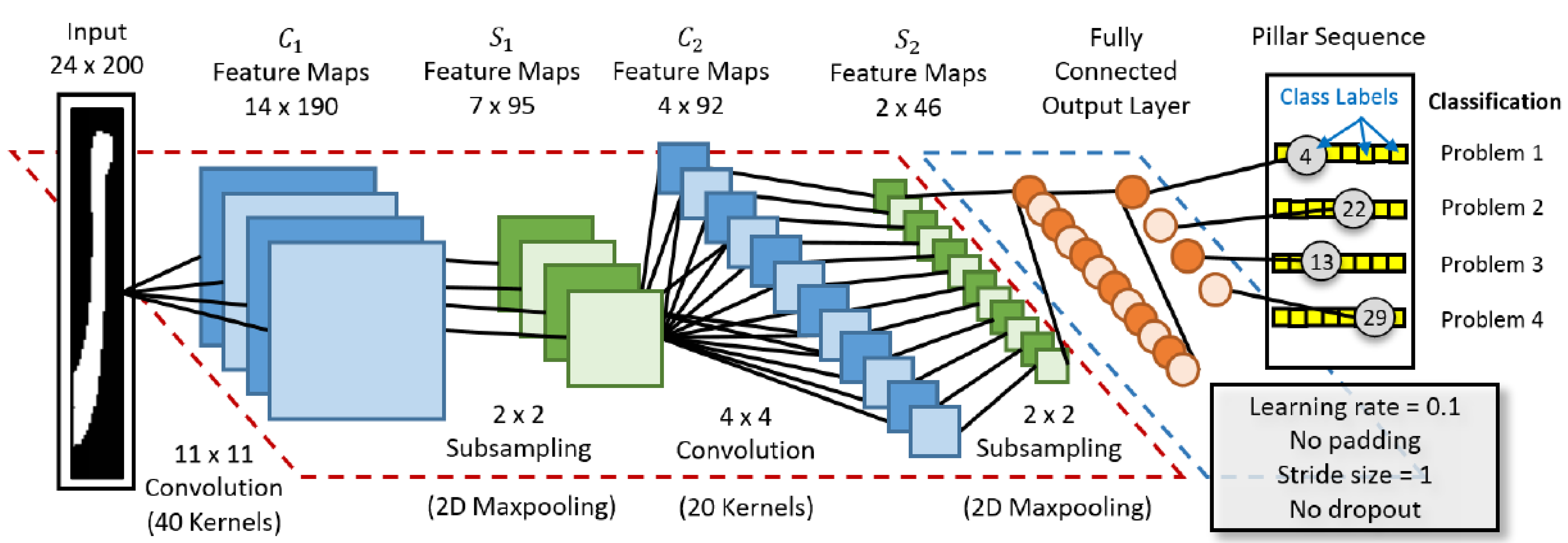


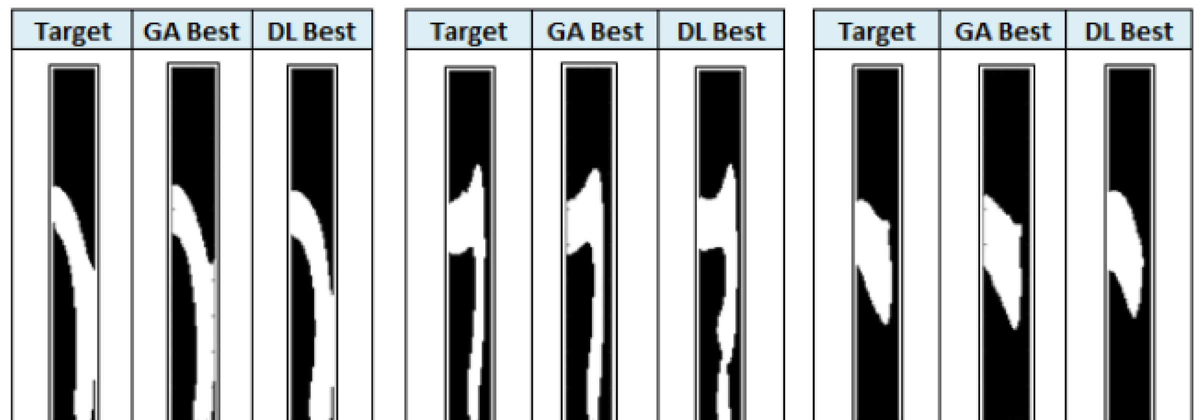
Fig 5: Implementation of the problem in the convolutional neural network with simultaneous multi-class classification (CNN-SMC)

Pixel Match Rate (PMR)	
Min	52.83%
Mean	88.09%
Median	89.92%
Max	100.00%
Percentage of Test Samples Exceeding PMR Threshold	
PMR >= 80 %	84.33%
PMR >= 85%	70.89%
PMR >= 90%	49.50%
PMR >= 95%	18.47%

Table 1: Prediction accuracy using current approach

PMR (%)	CNN-& (Random)	CNN-7 (Sobol)
Min	44.62	**50.36**
Mean	76.97	**78.44**
Median	77.46	**79.44**
Max	**98.11**	97.58

Table 2: Random vs Quasi-random training data



Target	GA Best	DL Best	Target	GA Best	DL Best	Target	GA Best	DL Best
PMR	98.79%	95.88%	PMR	97.60%	93.15%	PMR	98.63%	95.81%
Time (s)	11,222.3	11.3	Time (s)	11,654.6	11.3	Time (s)	12,198.0	11.3

Table 3: Three comparison example of performance between DL-based tools and genetic algorithm (GA)-based methods

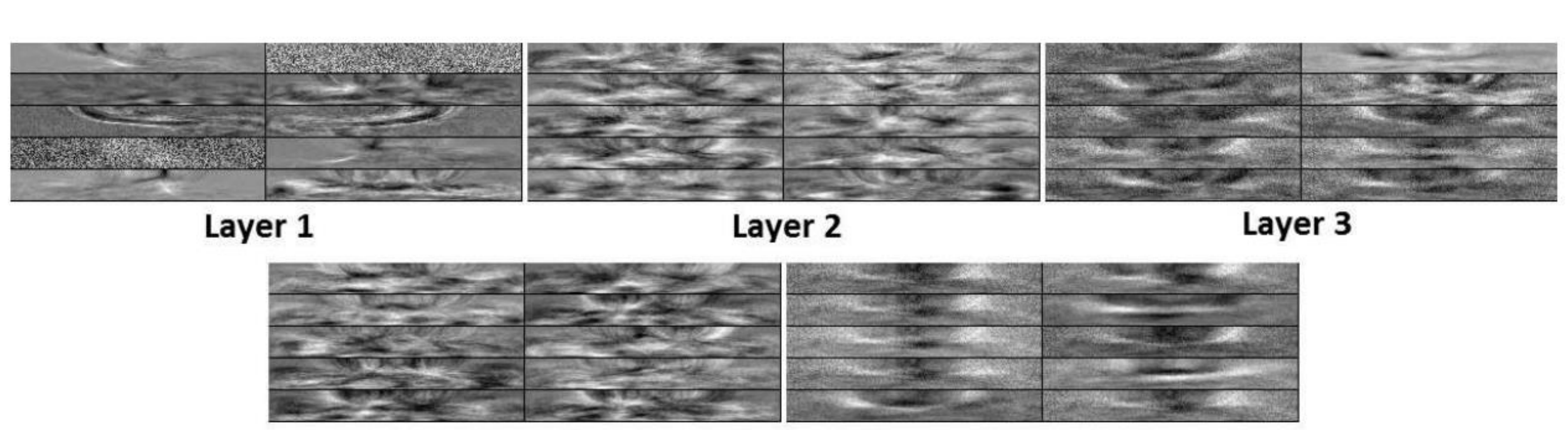


Fig 8: Weights of intermediate DNN layers

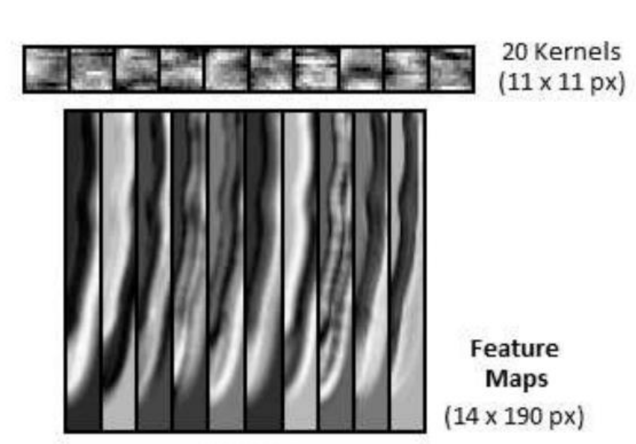


Fig 9: Filters and feature maps from the first convolutional layer

- **Pixel Match Rate (PMR):** percentage of pixels matching in color between the desired vs. reconstructed flow shape based on the predicted pillar sequence
- Current state-of-the-art genetic algorithm **(GA)-based methods perform slightly better quantitatively**, but suffers from **long execution time**.
- DL methods can achieve acceptable accuracy (Table 1), and even better with quasi-randomly generated data. (Table 2).
- DL methods **expedite the design process by more than 600 times** compared to GA and present real-time design alternatives for lab-on-a-chip design (Table 3).
- Intermediate layer weights offer insights to the diversity of flow shapes achievable through different pillar sequences. Fig 8 shows features from DNN architecture and Fig 9 shows filters from the CNN architecture.
- Framework solves the inverse problem without full-scale Navier-Stokes simulations in the **order of mere seconds**, allowing **for real-time design**

Future Work

- **Technical:** (1) Implementation of parallel processing on multiple GPUs to reduce training time with large volume of data, (2) optimization of network hyper-parameters, (3) investigation into other network types, (4) reframing class labels to decouple position and diameter
- **Analysis:** (1) Studying effects of using Sobol sequences on represented and unrepresented test data

References

[1] Hamed Amini, Elodie Sollier, Mahdokht Masaeli, Yu Xie, Baskar Ganapathysubramanian, Howard A. Stone, and Dino Di Carlo. Engineering fluid flow using sequenced microstructures. *Nature Communications*, 2013.

[2] Daniel Stoecklein, Chueh-Yu Wu, Keegan Owsley, Yu Xie, Dino Di Carlo, and Baskar Ganapathysubramanian. Micropillar sequence design for fundamental inertial flow transformations. *Lab on a Chip*, 2014.

[3] Daniel Stoecklein, Chueh-Yu Wu, Donghyuk Kim, Dino Di Carlo, and Baskar Ganapathysubramanian. Optimization of micropillar sequences for fluid flow sculpting. *arXiv preprint arXiv:1506:01111*, June 2015.

[4] Ilya M. Sobol, Danil Asotsky, Alexander Kreinin, and Sergei Kucherenko. Construction and comparison of high-dimensional sobol generators. *Wilmott Journal*, page 6479, 2011.