Architectures and Algorithms for User Customization of CNNs

Barend Harris, Mansureh S. Moghaddam, Duseok Kang, Inpyo Bae, Euiseok Kim,
Hyemi Min, Hansu Cho, Sukjin Kim, *Bernhard Egger*, Soonhoi Ha, Kiyoung Choi

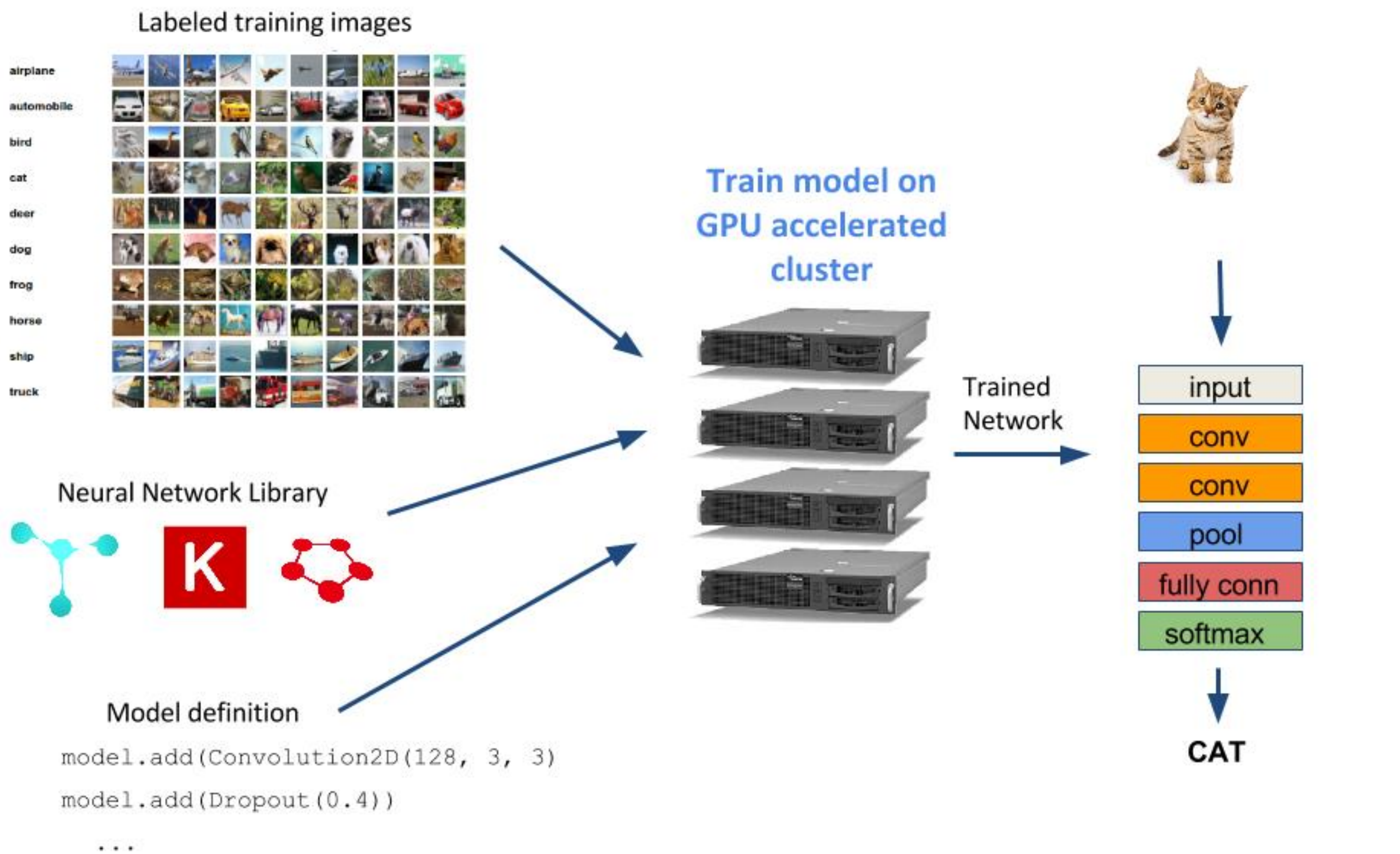
ASP-DAC 2018

ENGINEERING
COLLEGE OF ENGINEERING
SEOUL NATIONAL UNIVERSITY
서 울 대 학 교 공 과 대 학

SAMSUNG

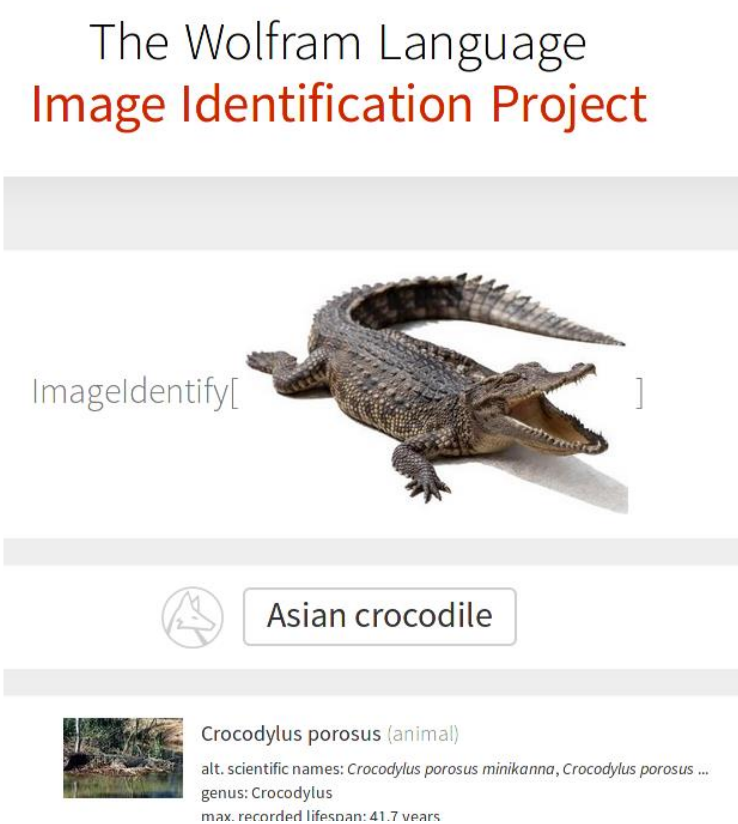
Classification with Convolutional Neural Networks (CNNs)

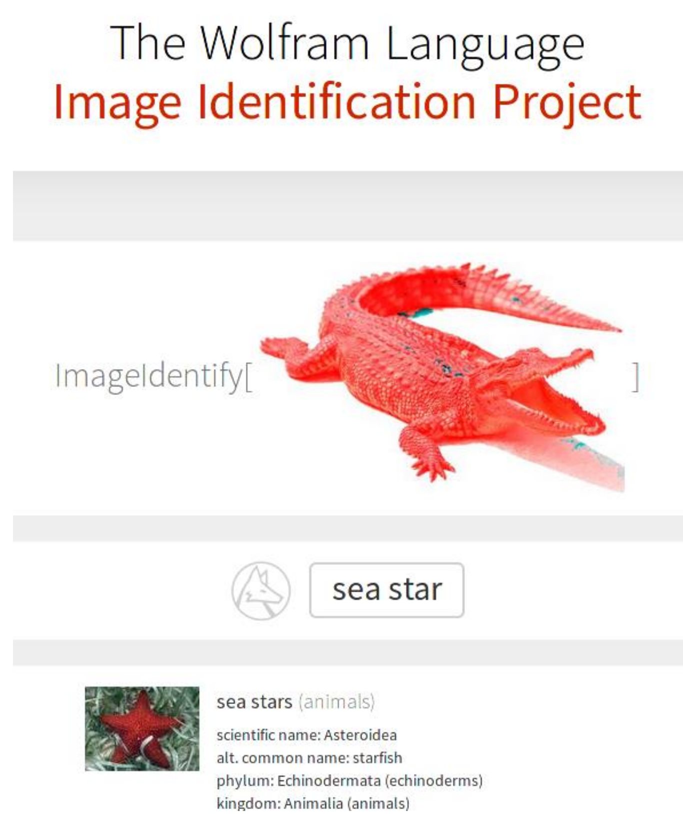
- **CNNs have shown excellent performance in many domains**
  - deep CNN + big data set + (computationally intensive) training



Classification with Convolutional Neural Networks (CNNs)

- **Large models have some inherent limitations**
  - difficult to obtain representative training data
  - the real world scenario may differ from the training scenario





Classification with Convolutional Neural Networks (CNNs)

- **We propose:**

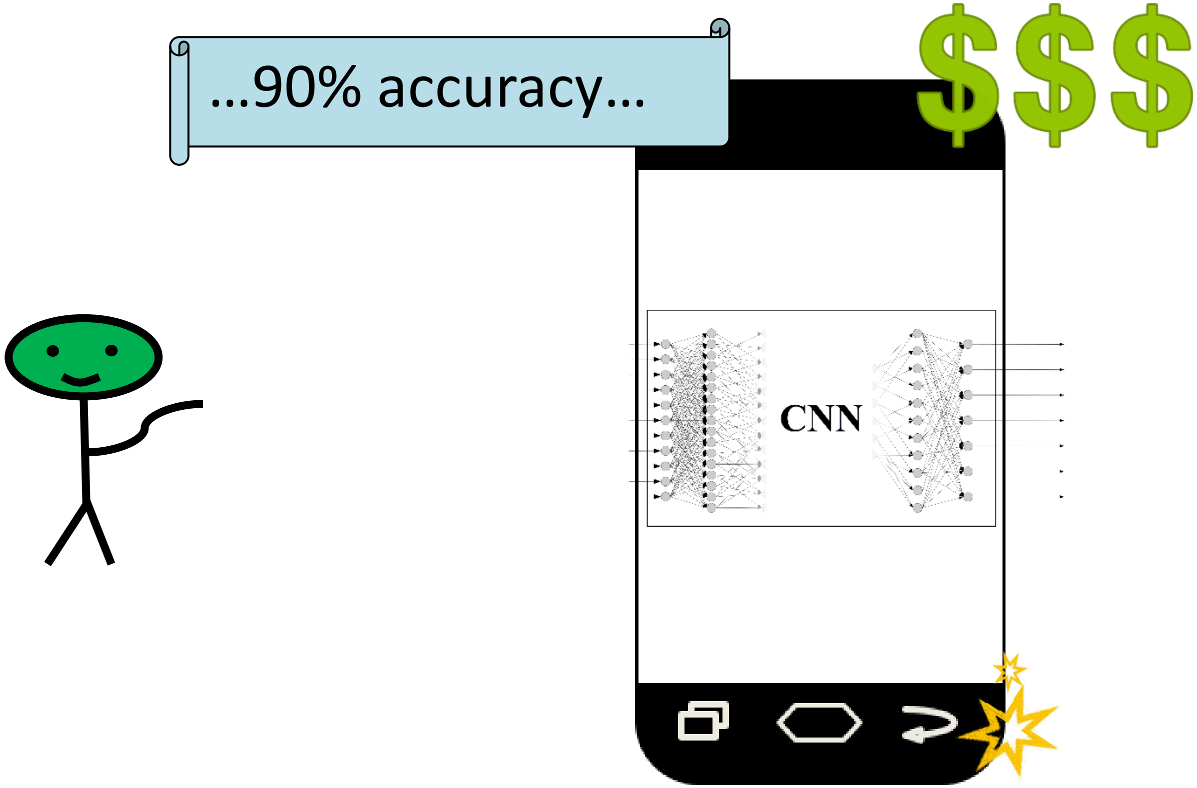
  A computationally light technique to increase the accuracy of a large general model using a small amount of on-device retraining

Consider Handwriting Recognition

- **CNN-based models achieve good accuracy on big data**
  - around 90% on NIST[1)] (a-zA-Z0-9) for train/test data set splits obtained from many individuals

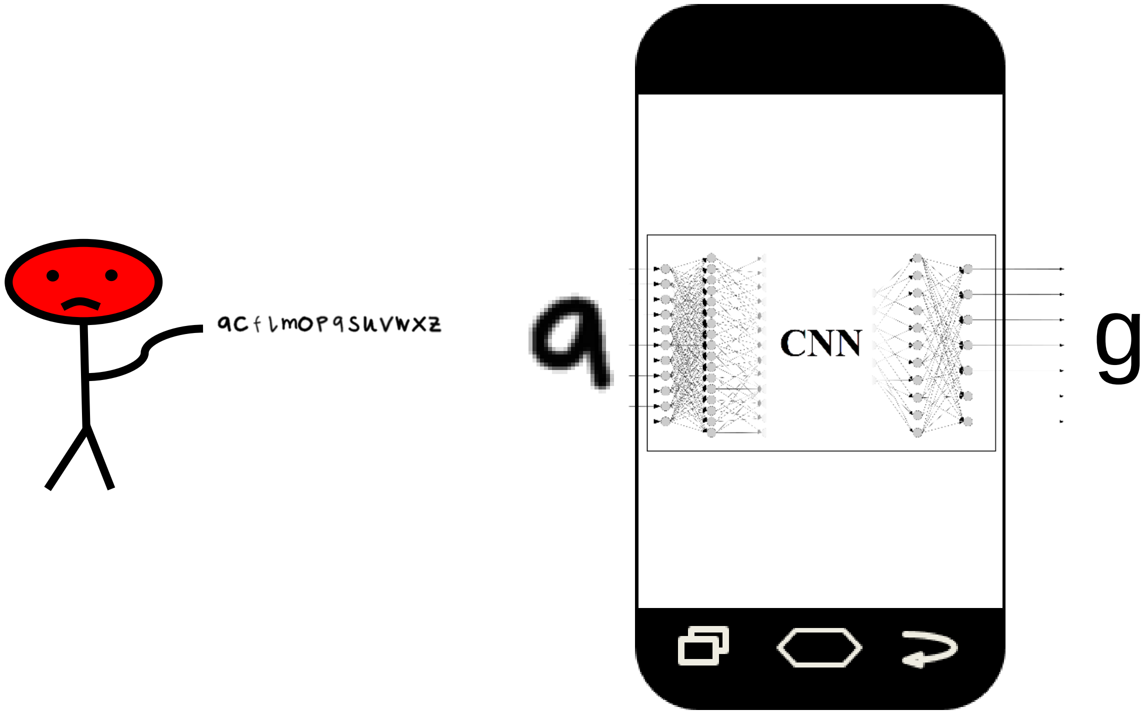
1) Ciresan et al. Convolutional neural network committees for handwritten character classification. International Conference on Document Analysis and Recognition, 2011

Consider Handwriting Recognition



- **Trained models are integrated into end-user devices**

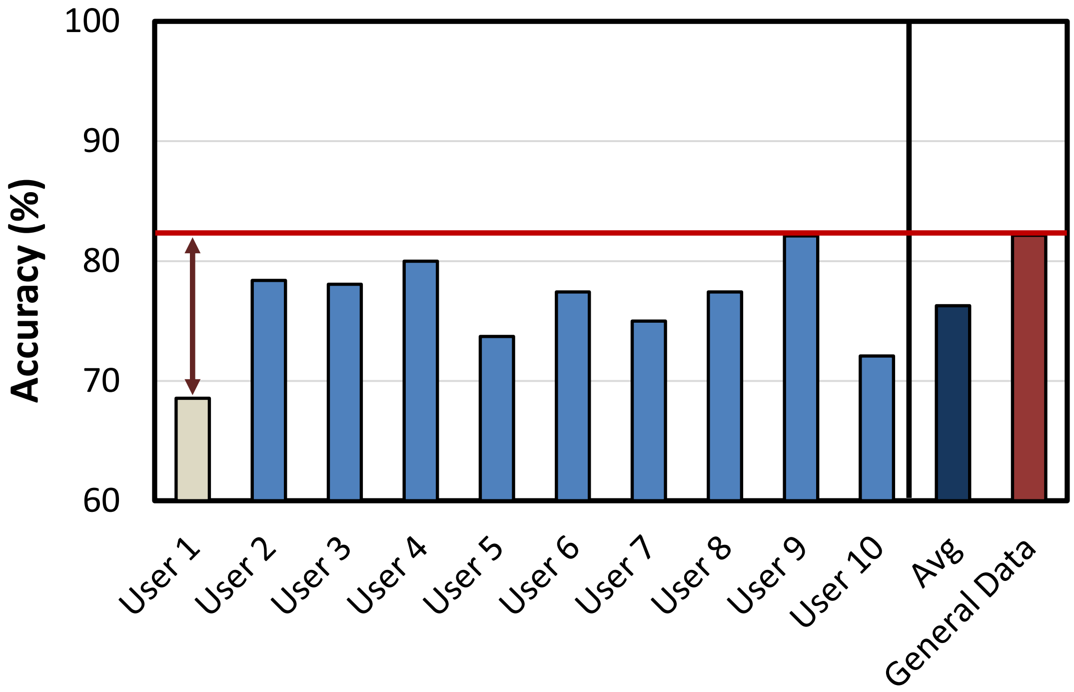
Consider Handwriting Recognition



- Performance for particular individual users not stellar

Handwriting Recognition

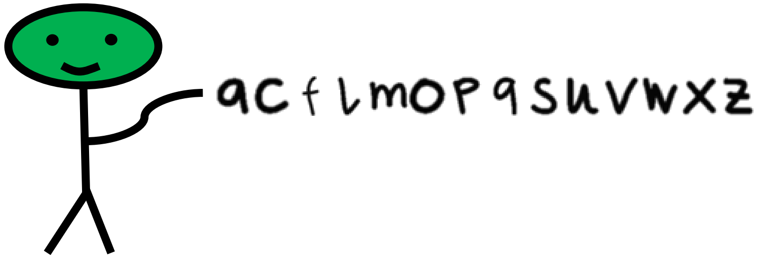
- **General Model (adapted LeNet-5[2]) trained with NIST database[3])**
  - **82%** general test set accuracy
  - **76%** when tested against individual user data



2) Lecun et al. Gradient-based learning applied to document recognition. IEEE, 1998
3) P.J. Grother. NIST special database 19 handprinted forms and characters database. National Institute of Standards and Technology, 2016.

On-Device Retraining is Challenging

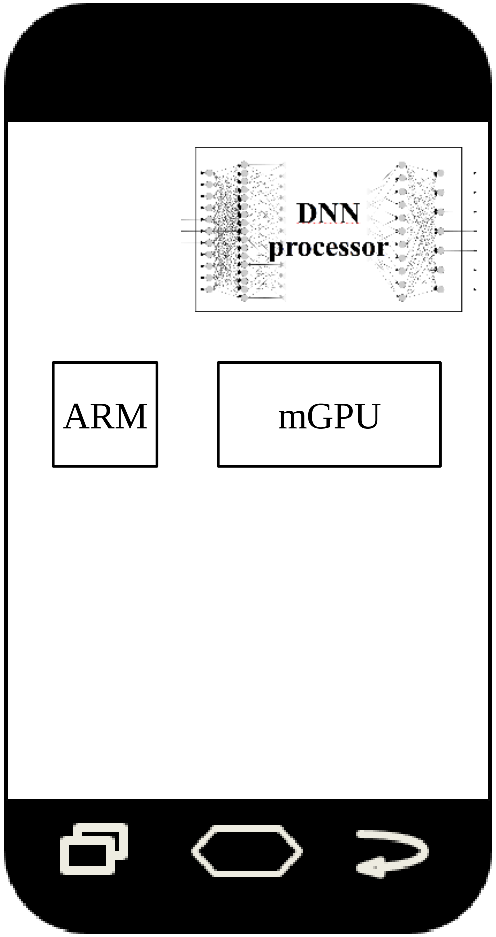
- **Energy, privacy, data size, catastrophic forgetting**



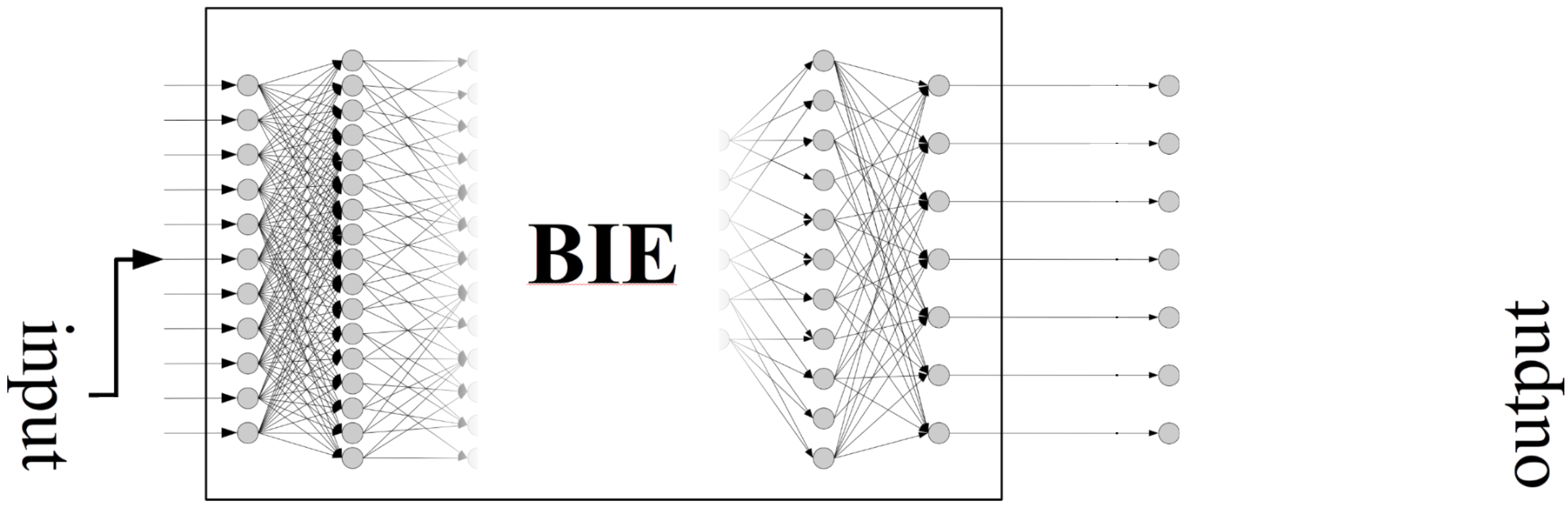
Methodology

Target Device

- **Mobile end-user device has**
  - a general purpose processor
  - a DNN processor
  - additional accelerators (mobile GPU, etc.)

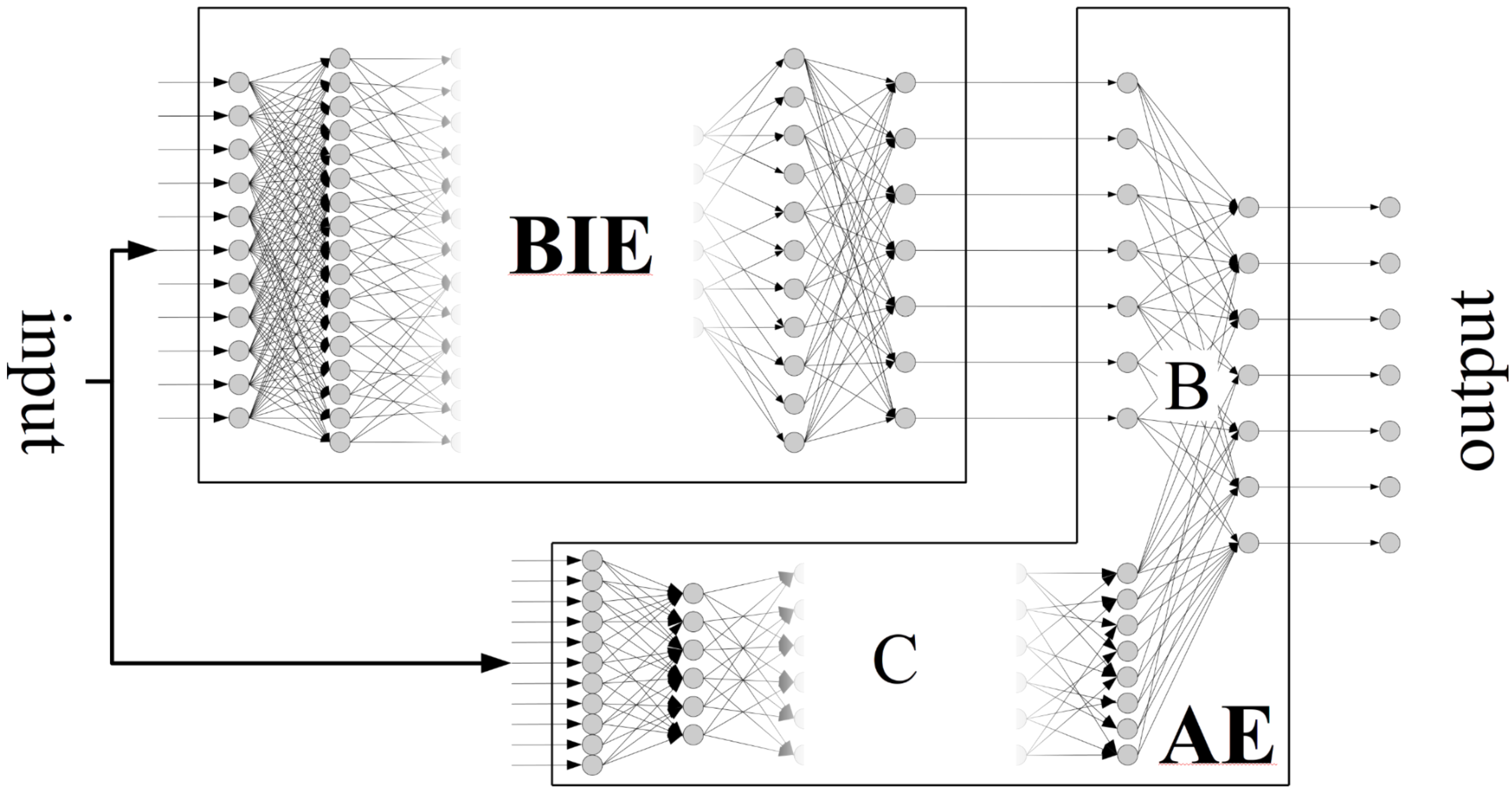


Proposed Network Architecture



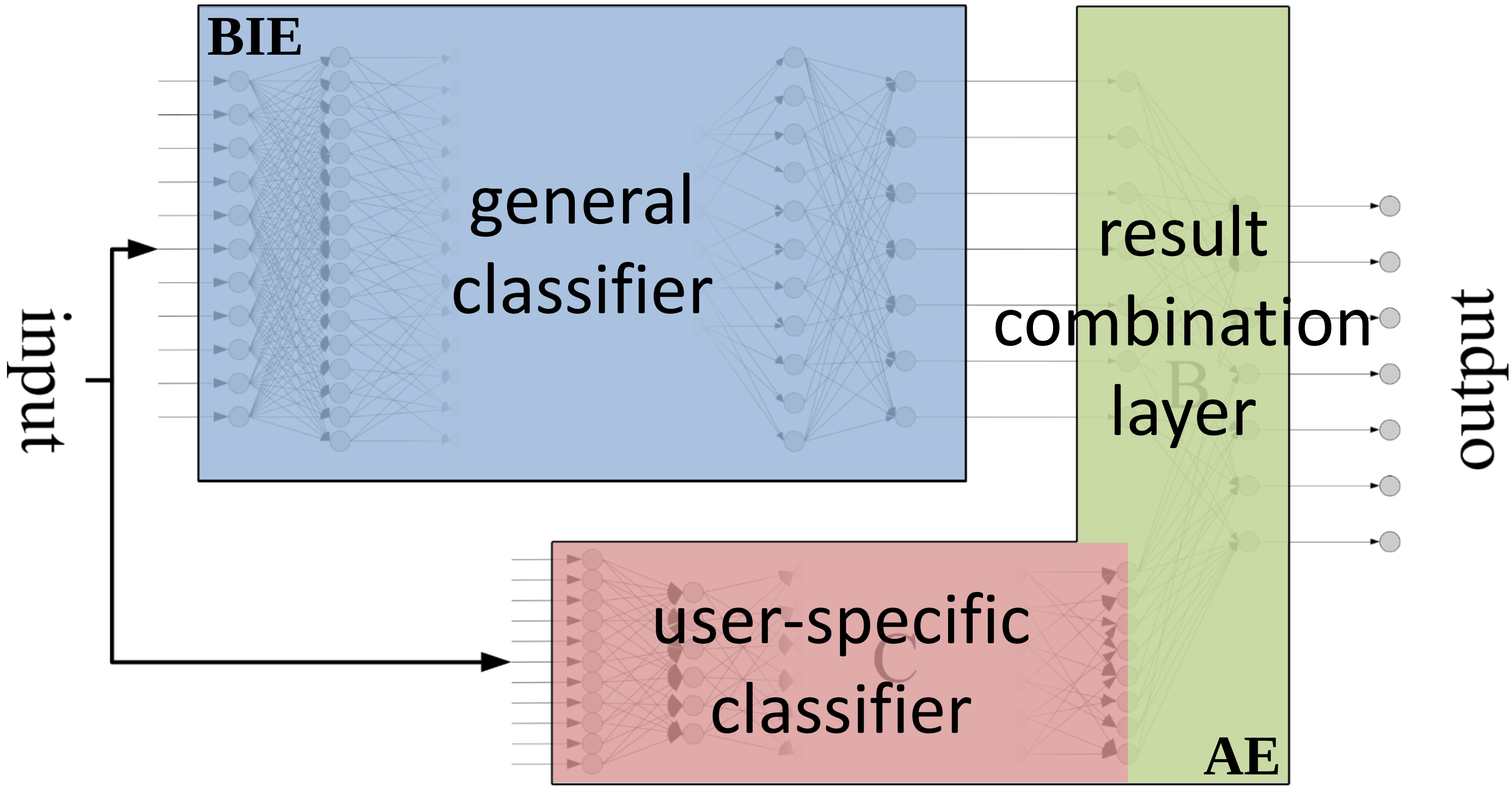
- **Basic Inference Engine (BIE)**
  - a **large CNN** accelerated with a **high power** dedicated accelerator
    - e.g. dedicated NN processor, FPGA, ASIC processor, mobile GPU

Proposed Network Architecture

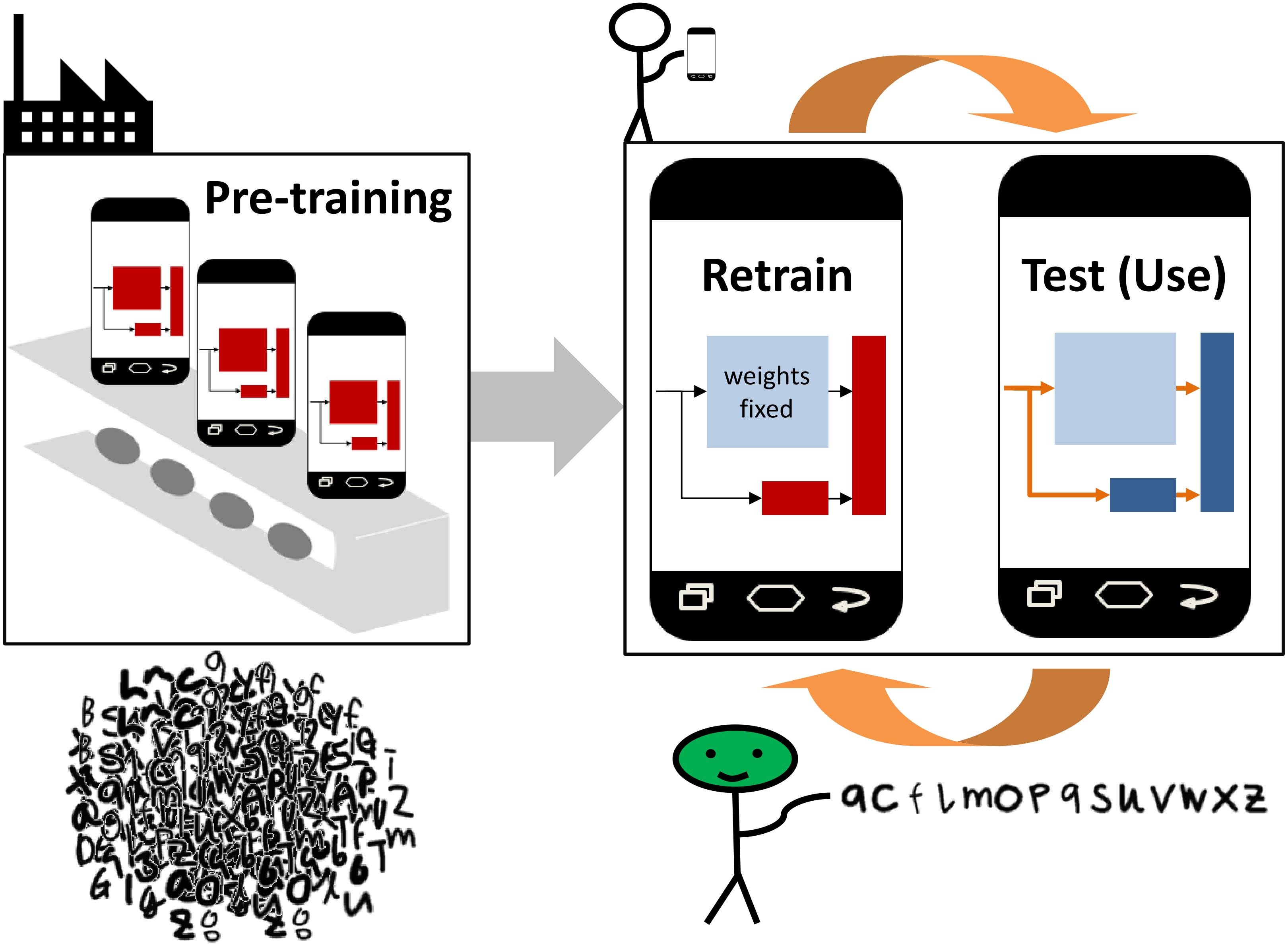


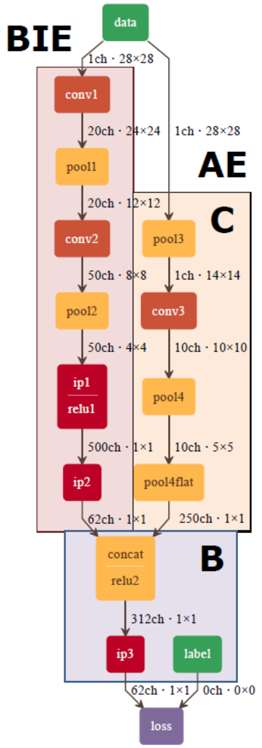
- **Augmenting Engine (AE)**
  - a **small CNN** accelerated with a **low power** general purpose accelerator

Proposed Network Architecture



Pre-Training and Retraining





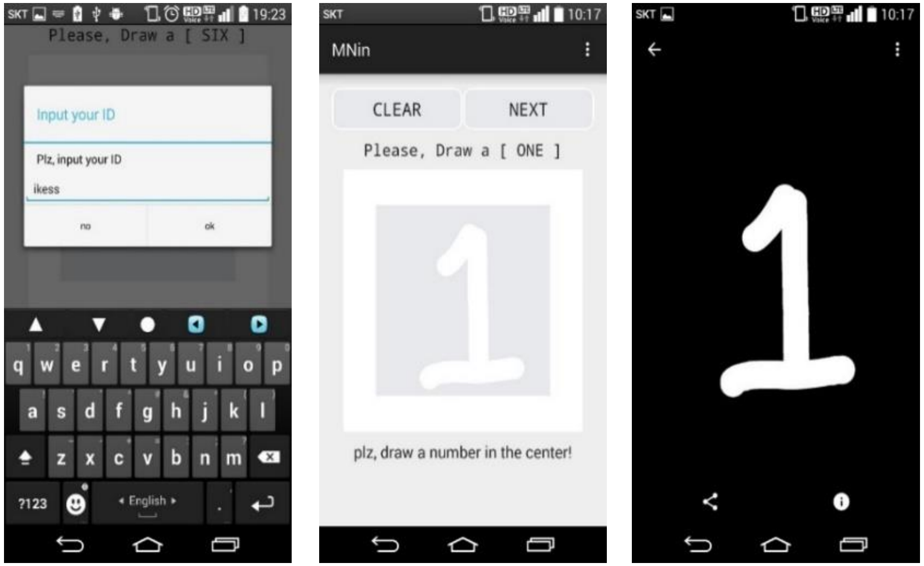
Application to Handwriting Recognition

Task: Handwritten Character Recognition

- **Handwritten Character Recognition**
  - 62 classes (a-z, A-Z, 0-9)

- **General Data:**
  - NIST Special Database 19

- **User Data:**
  - gathered with custom Android App



Handwritten Character Architecture

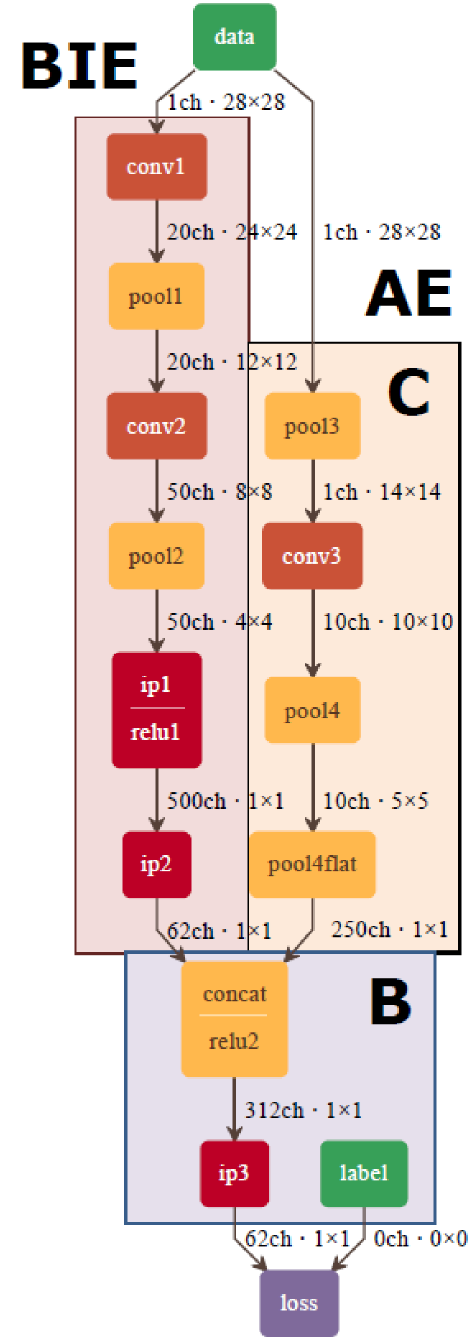
- **BIE**: LeNet-5
- **AE**: Small Convolutional Network

Relative overhead of AE wrt BIE

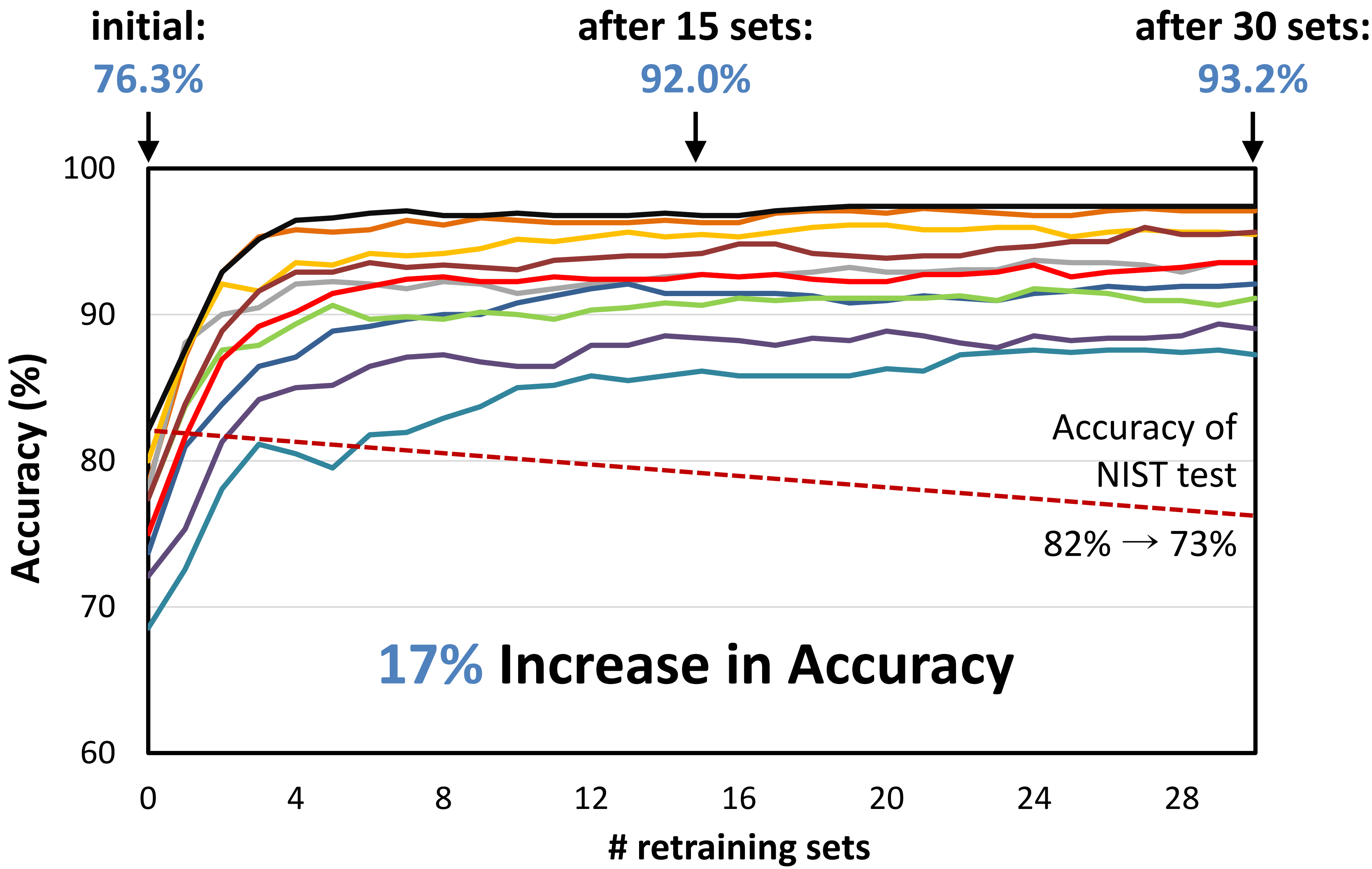
	MACC	#Neurons	#Weights
Inference	2%	12%	4%
Training	2%	12%	4%

Overhead of networks on one image

	MACC	#Neurons	#Weights
BIE Inference	2,319 k	79 kB	1,826 kB
AE Inference	44 k	9 kB	78 kB
BIE Training	4,167 k	155 kB	3,652 kB
AE Training	83 k	15 kB	157 kB

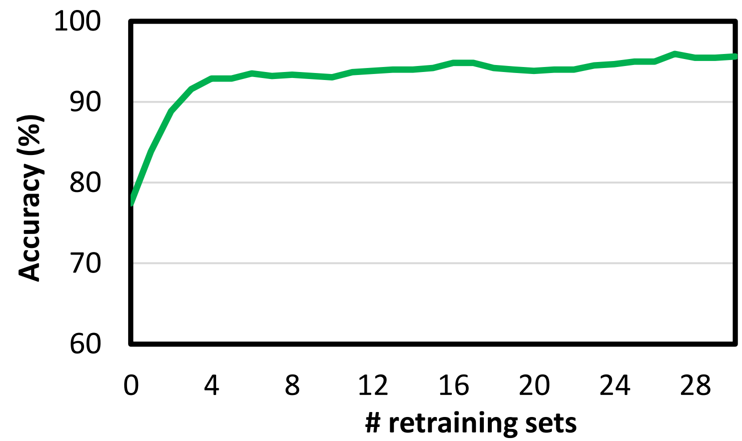


Results for 10 Individual Users



1 retraining set = 10 epochs of 1 full data set (62 characters)

Results for One Individual User

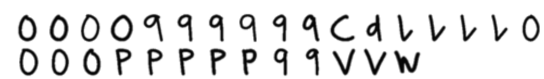


initial: 77.4%

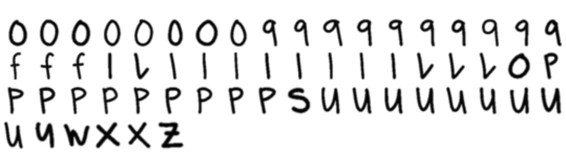
after 1 set: 86.1%

after 3 sets: 94.2%

after 10 sets: 95.2%

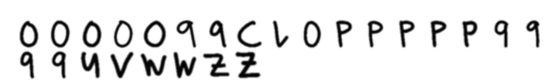


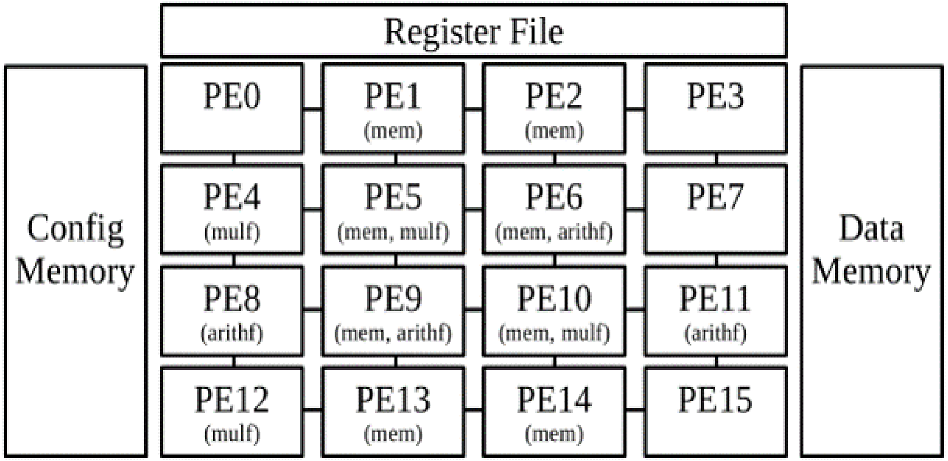
after 2 sets: 90.3%



after 5 sets: 92.6%

after 30 sets: 96.0%





Accelerating DNNs on a Modulo-Scheduled Coarse-Grained Reconfigurable Architecture

On Device Training with CGRAs

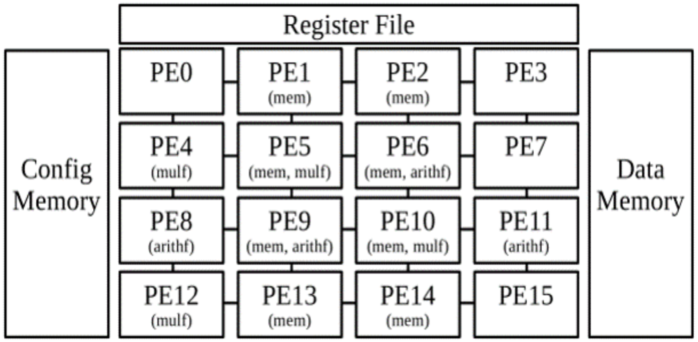
- **CGRAs are effective for CNN processing as:**
  - **CNN processing mainly consists of matrix multiplications**
    - contain many high iteration loops
  - **ideal for software pipelining**
    - can exploit loop level parallelism
    - hide memory latency

- **CGRA Data flow mode is similar to recently proposed CNN accelerator chips**

Adapted CGRA Architecture

- **We specifically use an adapted Samsung Reconfigurable Processor (SRP)**
  - already used in Samsung smartphones, TV's, printers and cameras

- **Hybrid VLIW-CGRA architecture**
  - 4x4 grid of 16 PEs
  - 8 x memory ld/st
  - 4 x floating point add/mul
  - 320KB on-chip SRAM



- **Custom CGRA C compiler**
  - increase schedulable scope by loop unrolling, fusion & interchange

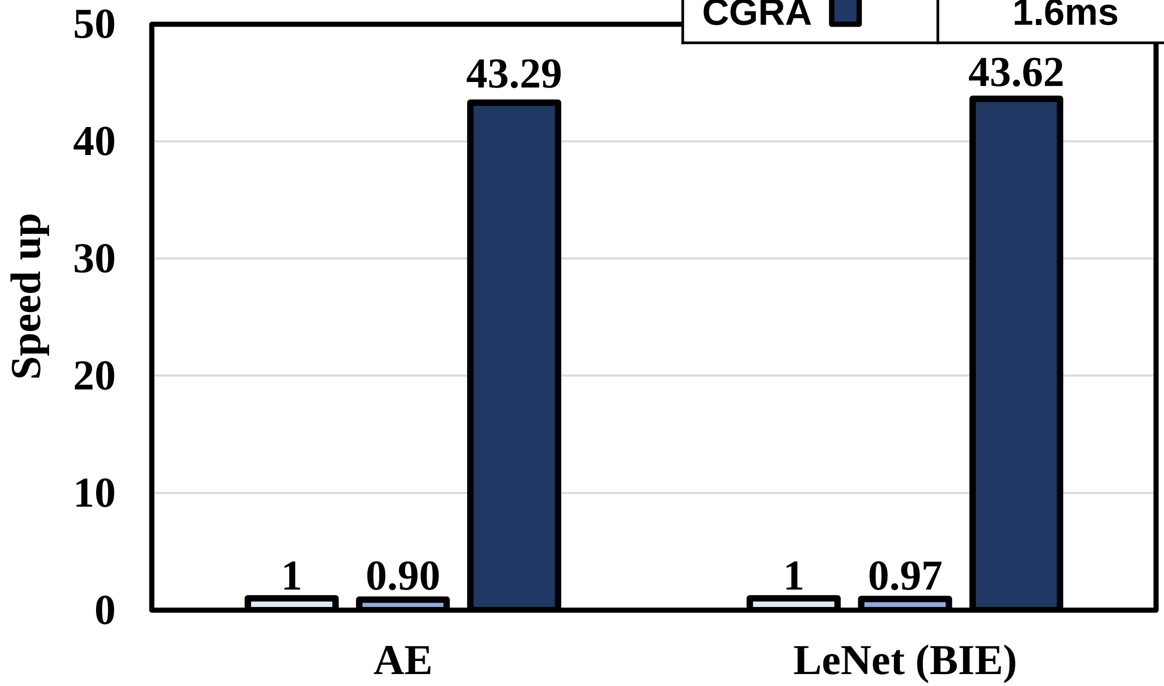
CGRA Acceleration results

- **Time taken to train on one image**
  - **45x speedup** compared to VLIW and ARMv7

ARM: v7 @1.2GHz
VLIW: 3-issue @500MHz
CGRA: 16 PEs @500MHz

Processing Time

	AE	LeNet (BIE)
ARM	70ms	2725ms
VLIW	78ms	2817ms
CGRA	1.6ms	62ms



CGRA power

- **Estimated power used for one image:**
  - **ARMv7:** **49 fold** power reduction
  - **VLIW:** **3 fold** power reduction

	Average Power [mW]	**Energy** [mJ]
ARMv7	169	11.83
VLIW	50	3.90
CGRA	150	0.24

Conclusion

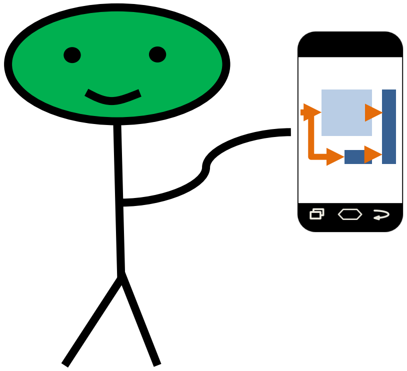
- **Adapting general CNN models to user data can increase the practicality of CNN applications**

- **BIE - AE architecture allows for:**
  - increased user accuracy
  - small training overhead
  - Handwritten Character Recognition:
    - **17%** accuracy increase
    - **76.8%** to **93.2%**

- **Can use CGRAs to accelerate on device training effectively**

Future Work

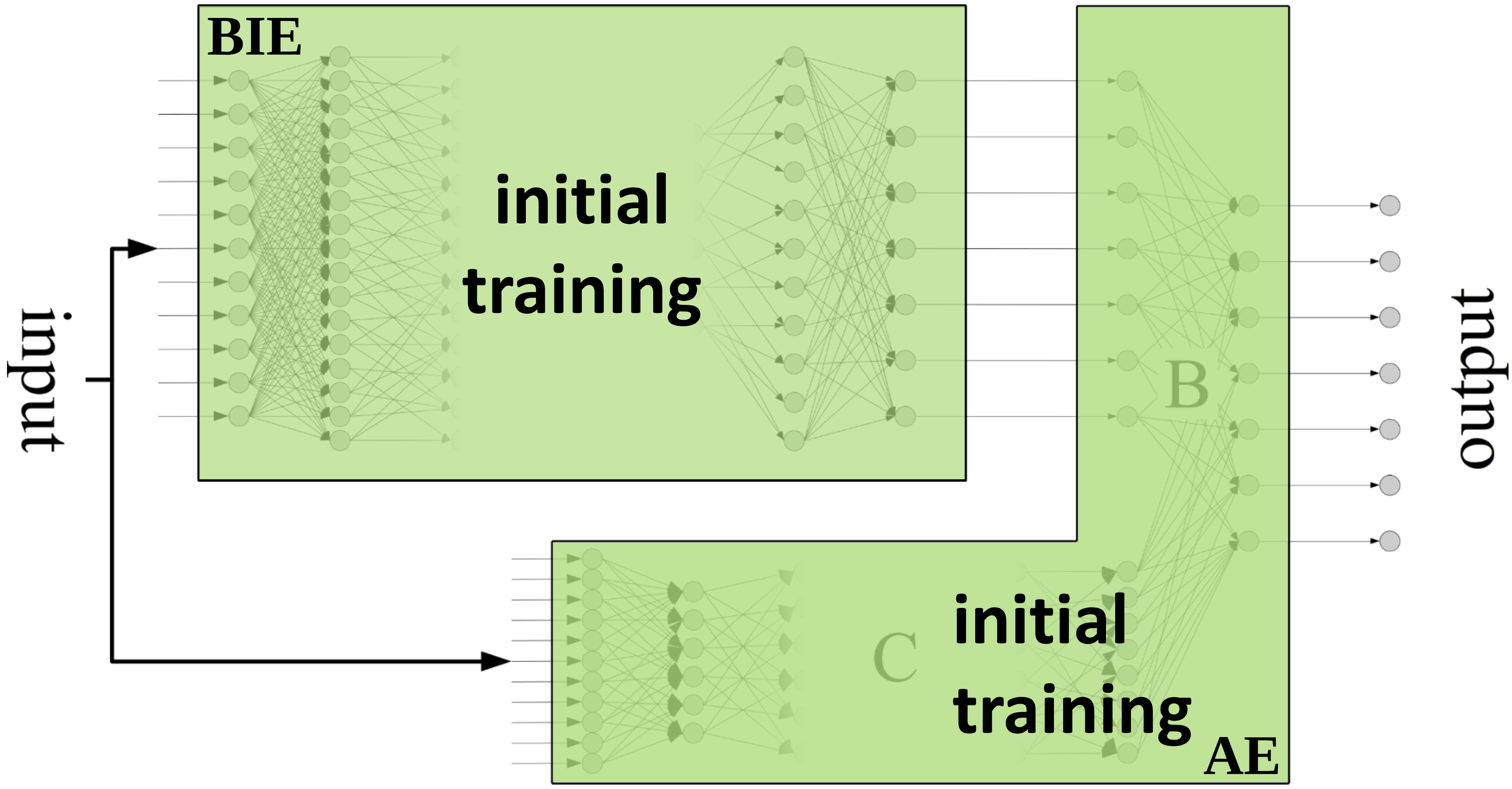
- **Apply to more complex problem domains**
  - **Hangul character classification**
    - 520 class task
    - 2350 class task
  - **Speech Recognition**



Thank you!

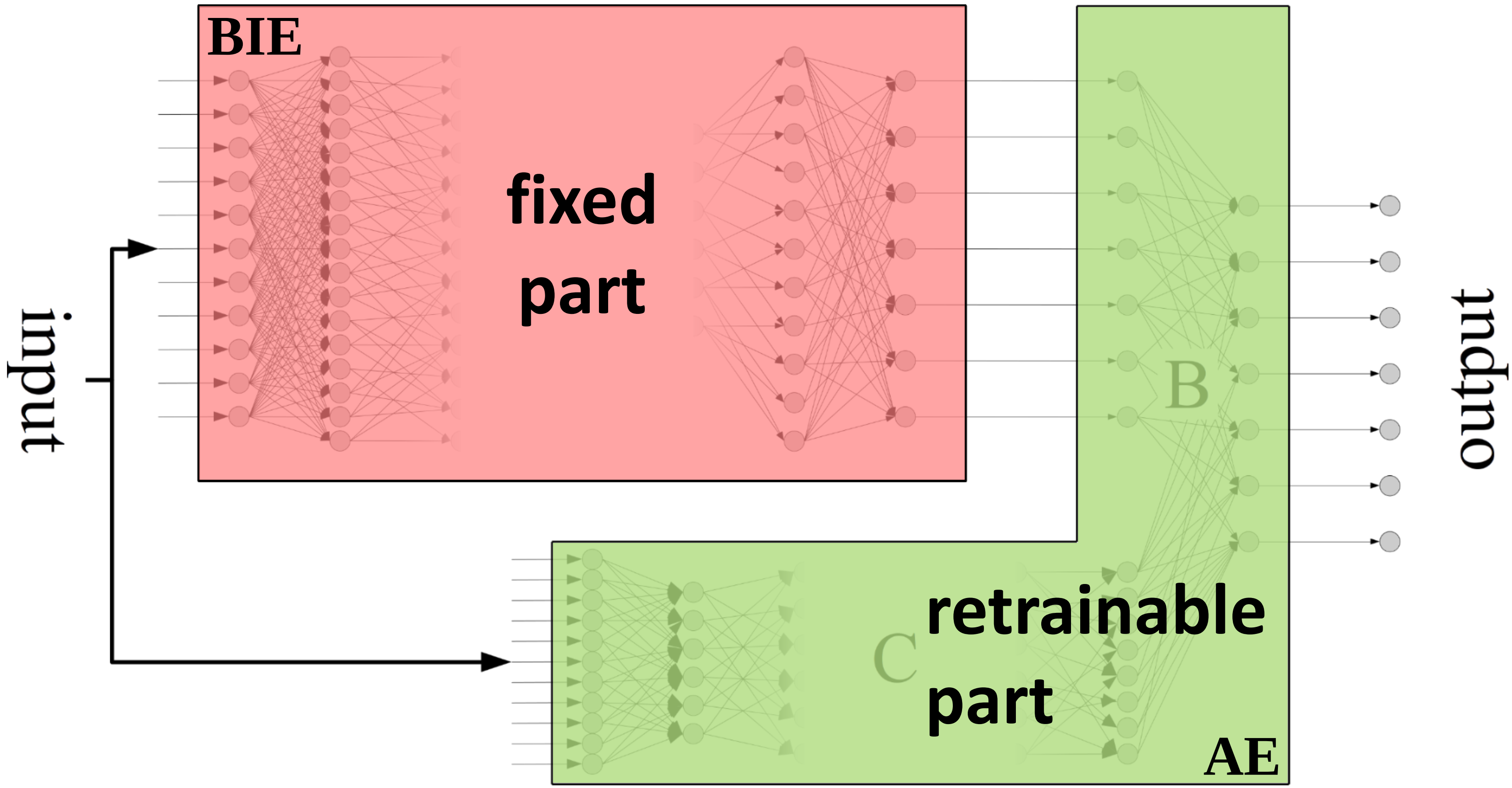
Proposed Network Architecture

- **Initial training**: retrain BIE + AE with big data

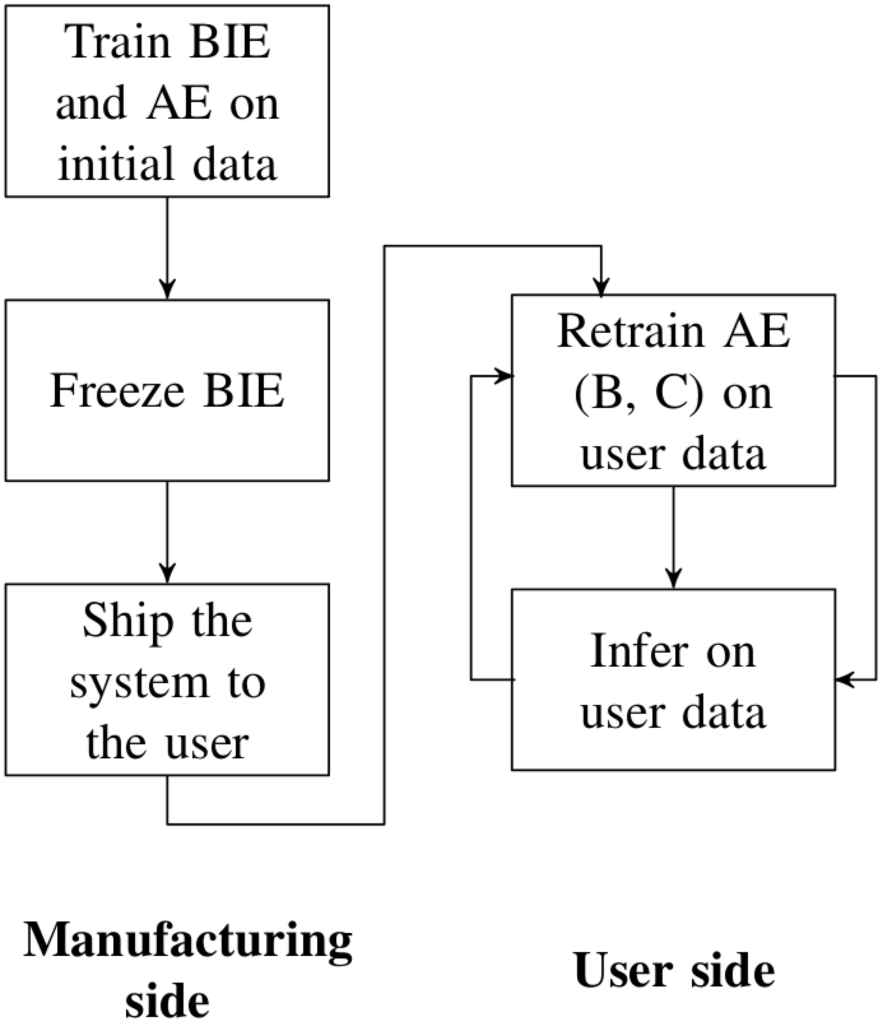


Proposed Network Architecture

- **On-device retraining**: retrain only AE using user-specific data



Training Flow



Detailed Results

Dataset	all		letter		lower		upper		digit	
	before	after	before	after	before	after	before	after	before	after
NIST	82.1	73.9	69.7	73.8	88.8	87.0	96.2	96.1	98.2	96.7
User 1	68.6	87.3	74.6	91.2	86.5	96.2	95.8	98.5	97.0	99.0
User 2	78.4	97.1	78.1	98.7	94.2	100	100	100	91.0	100
User 3	78.1	93.6	76.0	94.8	97.3	98.9	99.2	99.6	98.0	100
User 4	80.0	95.5	78.5	96.5	95.4	98.9	99.2	100	99.0	100
User 5	73.7	92.1	73.3	92.3	85.4	98.9	94.2	98.5	98.0	100
User 6	77.4	91.1	79.2	93.3	97.3	99.2	97.7	100	99.0	100
User 7	75.0	93.5	75.6	96.5	90.8	99.6	100	100	100	100
User 8	77.4	95.6	78.7	96.5	93.1	99.6	98.5	100	100	100
User 9	82.1	97.4	81.2	97.9	95.0	99.2	99.6	100	100	100
User 10	72.1	89.0	72.7	90.4	90.4	98.8	89.6	97.7	93.0	99.0
Average	76.3	**93.2**	76.8	**94.8**	92.5	**98.9**	97.4	**99.4**	97.5	**99.8**