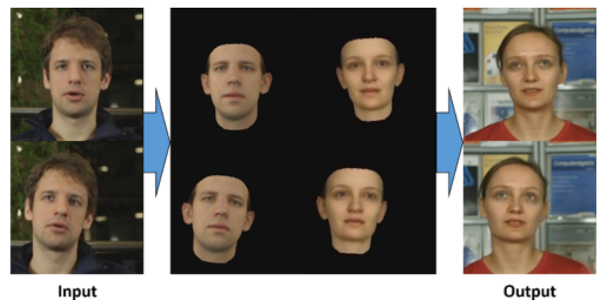
Capsule-Forensics: Using Capsule Networks to Detect Forged Images and Videos

Huy H. Nguyen (SOKENDAI, Japan) Junichi Yamagishi (NII, Japan) Isao Echizen (NII, Japan)

Generating of Fake Videos Impersonating a Person Using Deep Learning

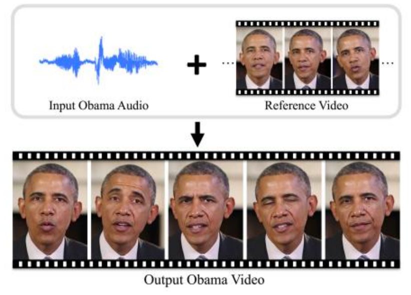
Face2Face: Real-time facial reenactment (Thies et al. 2016)



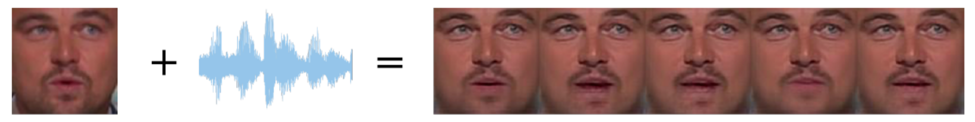
Deep Video Portraits: Face2Face + Translating head poses (Kim et al. 2018)

Deepfakes Video face swapping (2017)

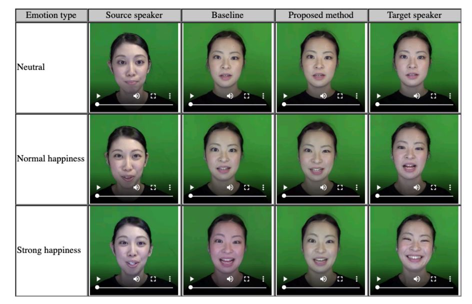
Bringing portraits to life (Averbuch-Elor et al. 2017)



Synthesizing Obama: Learning lip sync from audio (Suwajanakorn et al. 2017)



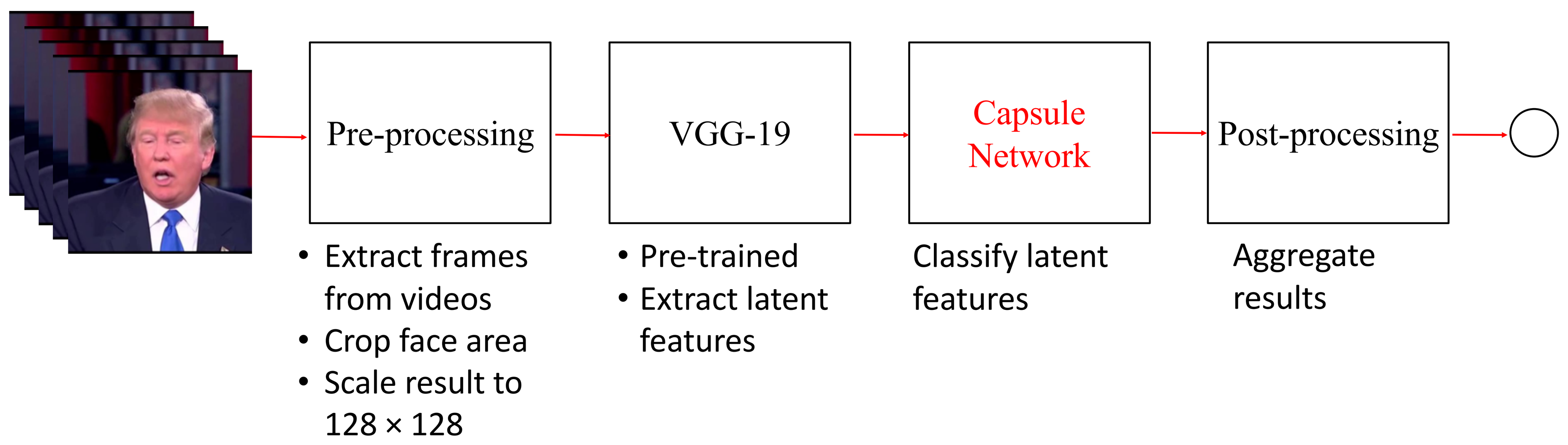
Speech2Vid (Chung et al. 2017)



Audiovisual speaker conversion (Fang et al. **ICASSP** 2019)

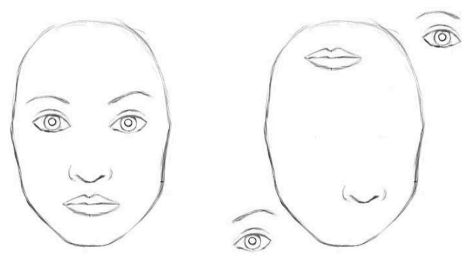
- Deep learning methods enable non-expert users with ordinary PCs to create realistic forged images and videos by using data available on social networks.
- Materials required for generating fake videos have been simplified over time.
- Forgery detectors need to be regularly updated to deal with
  - New kind of attacks
  - Better quality forged images/videos

→ Is there a general framework that could be applied for any kind of attack???



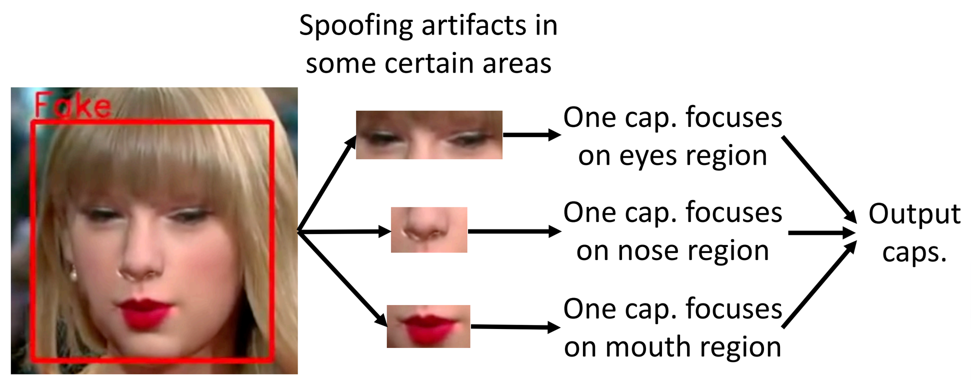
CNN vs. Capsule Network

- **In computer vision perspective**, convolutional neural networks (CNNs) has viewpoint invariant property but lacking of information about relative spatial relationships between features → Capsule network can solve this problem.

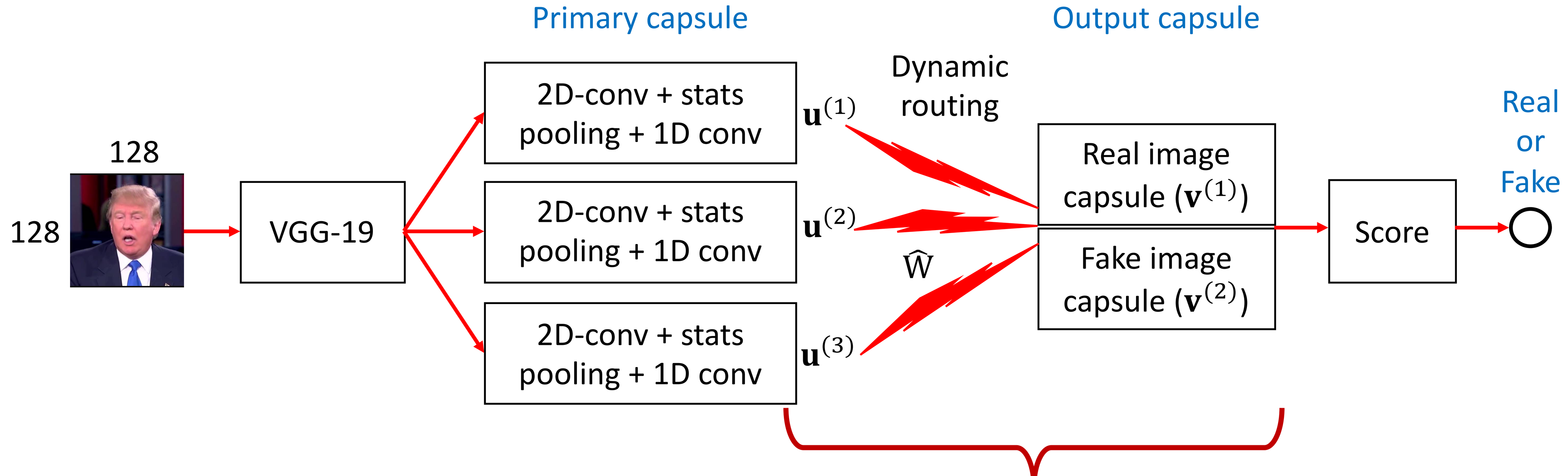


The two pictures are similar in the perspective of a CNN but dissimilar in the view of a capsule network.
Source: Max Pechyonkin.

- Capsule networks have several capsules, each capsule is a CNN learning some specific representations.
- The agreements between low-level capsules decides the activations of the high-level capsules.
- **In digital image forensics perspective**, each low-level capsule may capture some specific representations of spoofing artifacts in some certain area, or some specific kind of irregular noises created by presentation attacks.

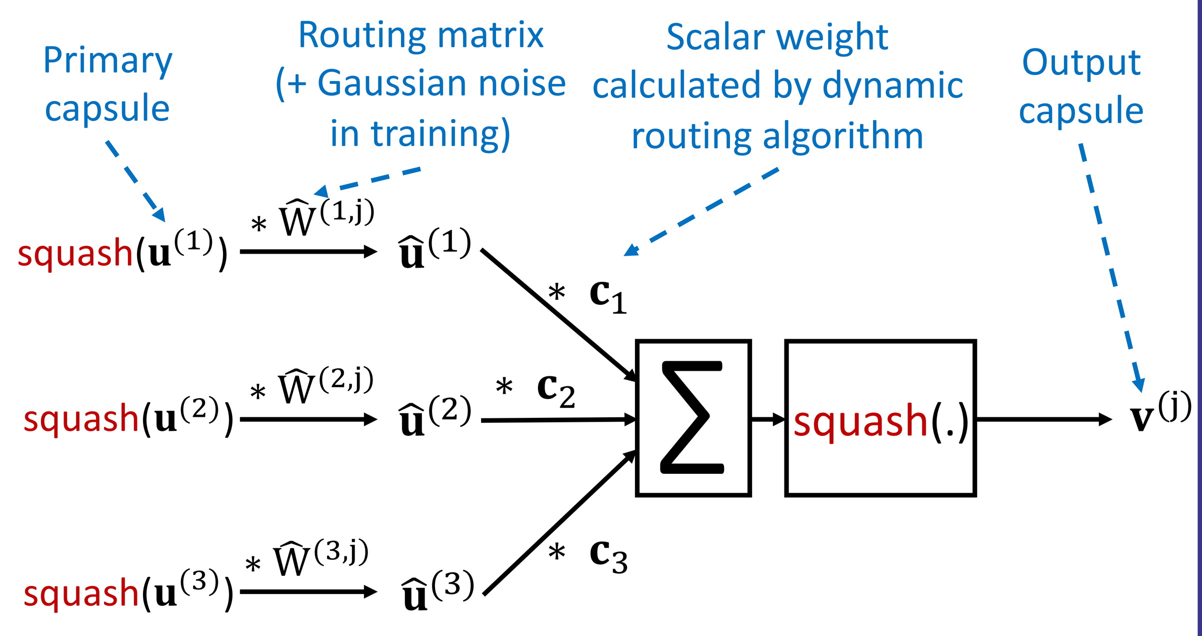


Capsule Network



Algorithm 1 Dynamic routing between capsules.

procedure ROUTING($\mathbf{u}^{(i)}, \mathrm{W}, r$)
  $\hat{\mathrm{W}} \leftarrow \mathrm{W} + rand(size(\mathrm{W}))$
  $\hat{\mathbf{u}}^{(i)} \leftarrow \hat{\mathrm{W}}^{(i,j)} squash(\mathbf{u}^{(i)})$
  for all input capsule i and all output capsules j **do**
    $b_{ij} \leftarrow 0$
  for r iterations **do**
    for all input capsules i **do** $\mathbf{c}_i \leftarrow softmax(\mathbf{b}_i)$
    for all output capsules j **do** $\mathbf{s}_j \leftarrow \sum_i c_{ij}\hat{\mathbf{u}}^{(i)}$
    for all output capsules j **do** $\mathbf{v}^{(j)} \leftarrow squash(\mathbf{s}_j)$
    for all input capsules i and output capsules j **do**
      $b_{ij} \leftarrow b_{ij} + \hat{\mathbf{u}}^{(i)} \cdot \mathbf{v}^{(j)}$
  return $\mathbf{v}^{(j)}$

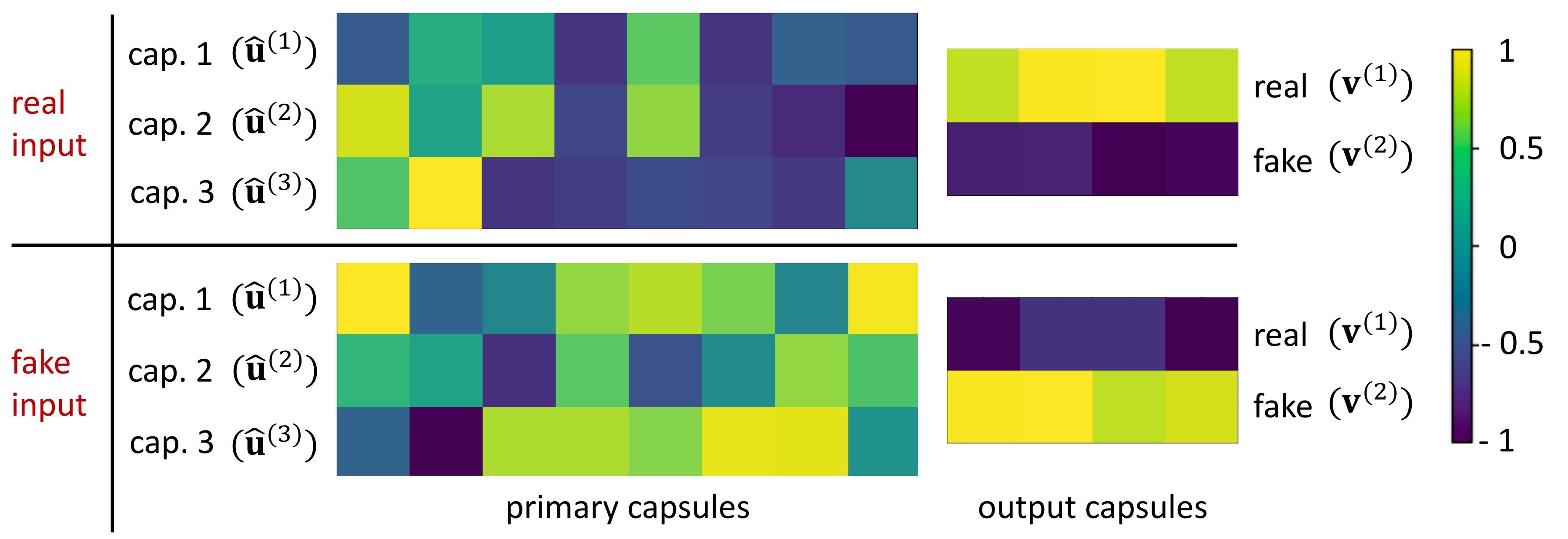


Squash function, used to scale vector magnitude to unit length:

$$\mathrm{squash}(\mathbf{u}) = \frac{\|\mathbf{u}\|_2^2}{1+\|\mathbf{u}\|_2^2}\frac{1}{\|\mathbf{u}\|_2}\mathbf{u}$$

Score function, used to determine the predicted label probabilities:

$$\hat{\mathbf{y}} = \frac{1}{m}\sum_i \mathrm{softmax}\left(\begin{bmatrix}\mathbf{v}^{(1)\mathsf{T}} \\ \mathbf{v}^{(2)\mathsf{T}}\end{bmatrix}_{:,i}\right)$$



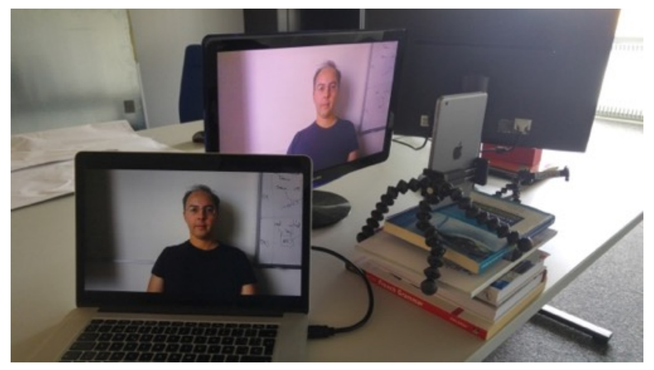
Visualization of activations of primary capsules ($\mathbf{u}^{(i)}$) and output capsules ($\mathbf{v}^{(j)}$)

Evaluation

Type of Attack	Detection Accuracy (%)
Replay Attack	100.00 *
CG vs. Natural Images	100.00 *
Deepfakes (Frames)	95.93 *
Deepfakes (Video)	99.23 *
Face2Face (c0 - Frames)	99.37
Face2Face (c0 - Video)	99.33
Face2Face (c23 - Frames)	96.50
Face2Face (c23 - Video)	96.00
Face2Face (c40 - Frames)	81.00
Face2Face (c40 - Video)	83.33

Note:
- c0, c23, c40: H264 compression levels
- number *: state-of-the-art result

Some examples from the evaluation datasets:



Idiap Replay Attack database (Chingovska et al. 2012)

Facial reenactment (Face2Face) in the FaceForensics database (Rössler et al. 2018)