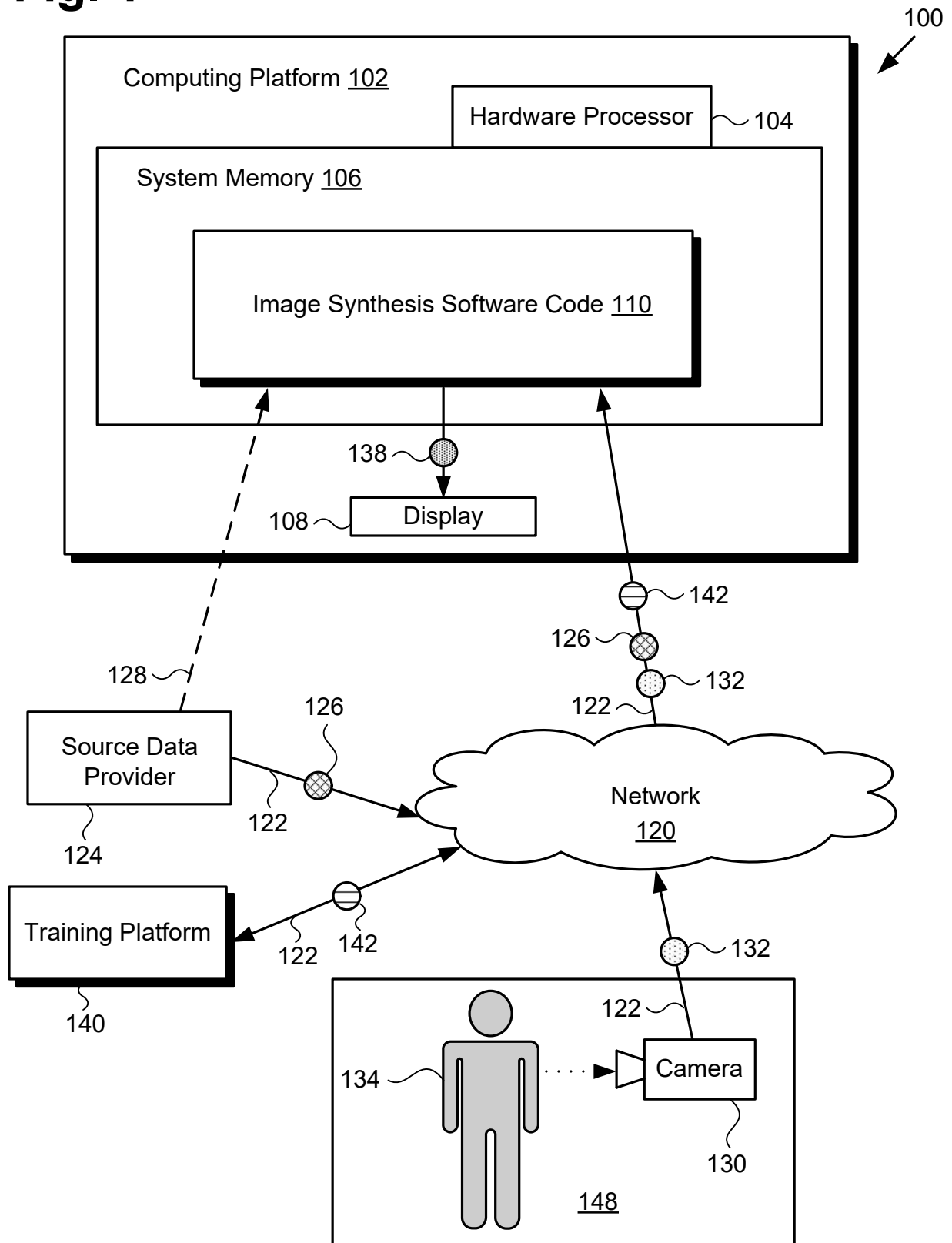




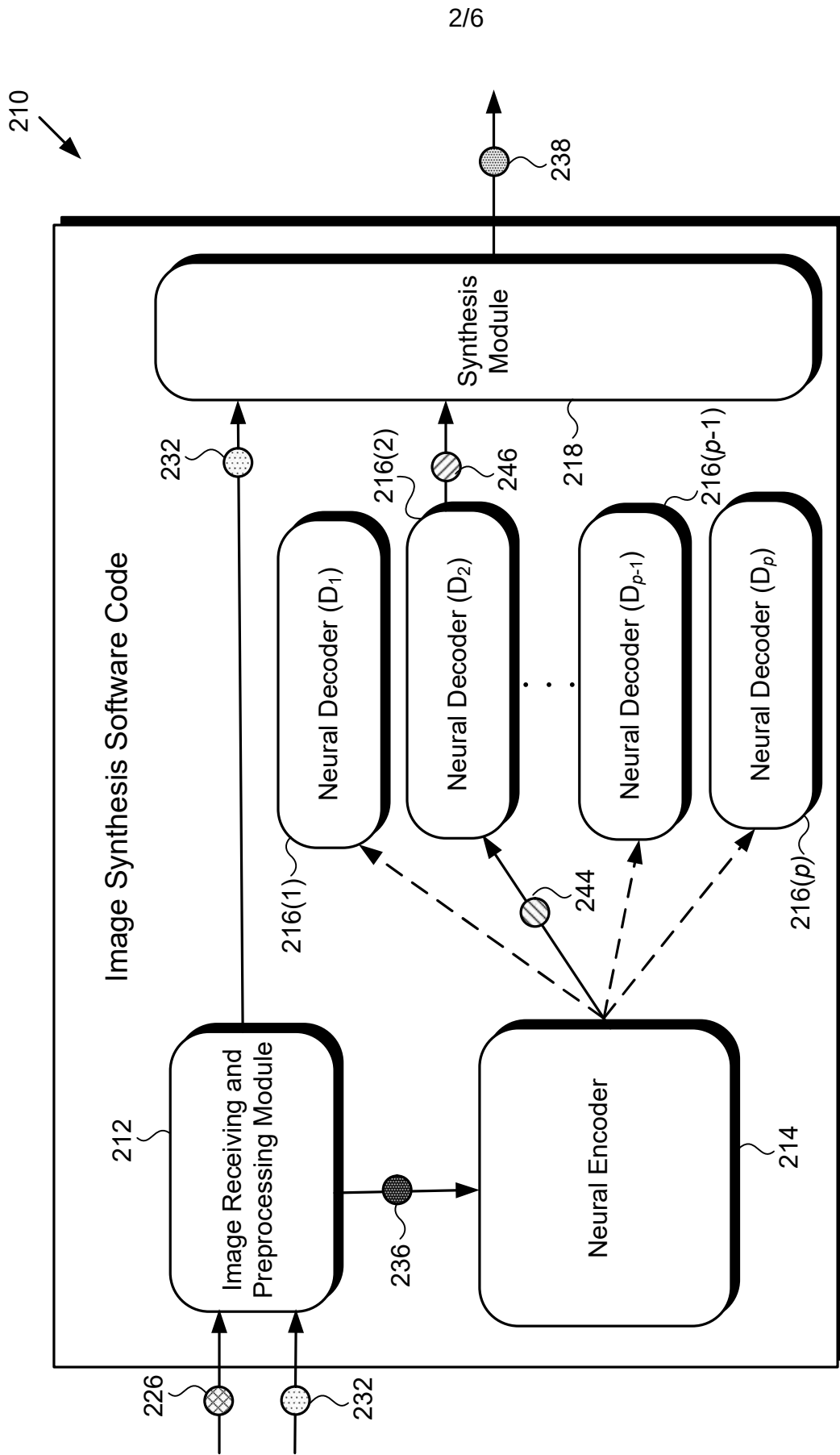


Fig. 2

Fig. 3A

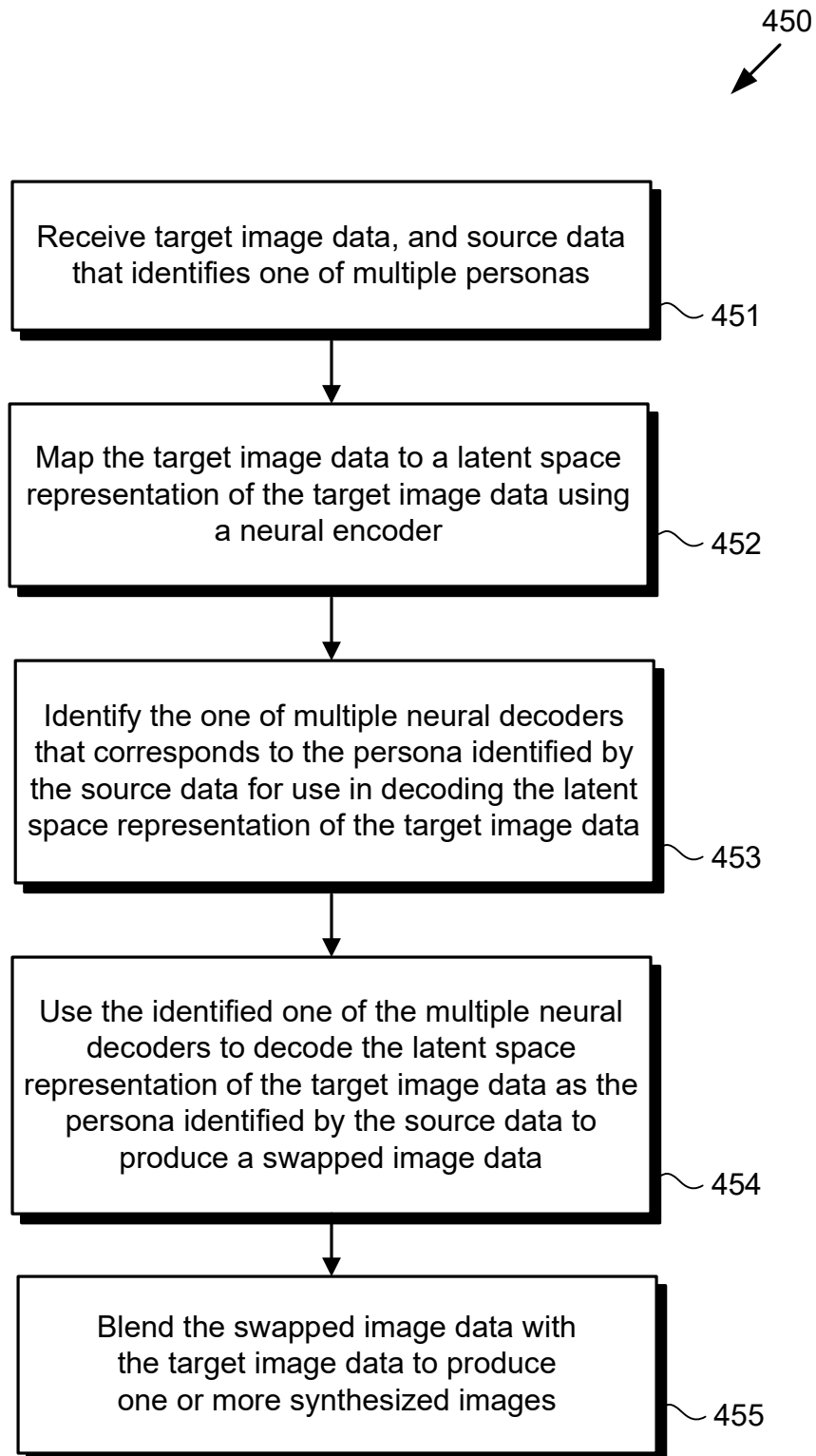
314

Lvl	Encoder	Activation	Output shape	Params
8	Input Image ~ 336	-	$3 \times 1024 \times 1024$	-
	Conv 1×1	LeakyReLU	$16 \times 1024 \times 1024$	64
	Conv 3×3	LeakyReLU	$16 \times 1024 \times 1024$	2.3k
	Conv 3×3	LeakyReLU	$32 \times 1024 \times 1024$	4.6k
	Downsample	-	$32 \times 512 \times 512$	-
7	Conv 3×3	LeakyReLU	$32 \times 512 \times 512$	9.2k
	Conv 3×3	LeakyReLU	$64 \times 512 \times 512$	18k
	Downsample	-	$64 \times 256 \times 256$	-
6	Conv 3×3	LeakyReLU	$32 \times 256 \times 256$	37k
	Conv 3×3	LeakyReLU	$128 \times 256 \times 256$	74k
	Downsample	-	$128 \times 128 \times 128$	-
5	Conv 3×3	LeakyReLU	$128 \times 128 \times 128$	148k
	Conv 3×3	LeakyReLU	$256 \times 128 \times 128$	295k
	Downsample	-	$256 \times 64 \times 64$	-
4	Conv 3×3	LeakyReLU	$256 \times 64 \times 64$	590k
	Conv 3×3	LeakyReLU	$512 \times 64 \times 64$	1.2M
	Downsample	-	$512 \times 32 \times 32$	-
3	Conv 3×3	LeakyReLU	$512 \times 32 \times 32$	2.4M
	Conv 3×3	LeakyReLU	$512 \times 32 \times 32$	2.4M
	Downsample	-	$512 \times 16 \times 16$	-
2	Conv 3×3	LeakyReLU	$512 \times 16 \times 16$	2.4M
	Conv 3×3	LeakyReLU	$512 \times 16 \times 16$	2.4M
	Downsample	-	$512 \times 8 \times 8$	-
1	Conv 3×3	LeakyReLU	$512 \times 8 \times 8$	2.4M
	Conv 3×3	LeakyReLU	$512 \times 8 \times 8$	2.4M
	Downsample	-	$512 \times 4 \times 4$	-
0	Conv 3×3	LeakyReLU	$512 \times 4 \times 4$	2.4M
	Conv 4×4	LeakyReLU	$512 \times 1 \times 1$	4.M
	Latent vector ~ 344	-	$512 \times 1 \times 1$	513
				23.1M

Fig. 3B

316

Lvl	Decoder	Activation	Output shape	Params
0	Latent vector ~ 344	-	$512 \times 1 \times 1$	-
	Conv 4×4	LeakyReLU	$512 \times 4 \times 4$	4.2M
	Conv 3×3	LeakyReLU	$512 \times 4 \times 4$	2.4M
1	Upsample	-	$512 \times 8 \times 8$	-
	Conv 3×3	LeakyReLU	$512 \times 8 \times 8$	2.4M
	Conv 3×3	LeakyReLU	$512 \times 8 \times 8$	2.4M
2	Upsample	-	$512 \times 16 \times 16$	-
	Conv 3×3	LeakyReLU	$512 \times 16 \times 16$	2.4M
	Conv 3×3	LeakyReLU	$512 \times 16 \times 16$	2.4M
3	Upsample	-	$512 \times 32 \times 32$	-
	Conv 3×3	LeakyReLU	$512 \times 32 \times 32$	2.4M
	Conv 3×3	LeakyReLU	$512 \times 32 \times 32$	2.4M
4	Upsample	-	$512 \times 64 \times 64$	-
	Conv 3×3	LeakyReLU	$256 \times 64 \times 64$	1.2M
	Conv 3×3	LeakyReLU	$256 \times 64 \times 64$	590k
5	Upsample	-	$256 \times 128 \times 128$	-
	Conv 3×3	LeakyReLU	$128 \times 128 \times 128$	295k
	Conv 3×3	LeakyReLU	$128 \times 128 \times 128$	148k
6	Upsample	-	$128 \times 256 \times 256$	-
	Conv 3×3	LeakyReLU	$64 \times 256 \times 256$	74k
	Conv 3×3	LeakyReLU	$64 \times 256 \times 256$	37k
7	Upsample	-	$64 \times 512 \times 512$	-
	Conv 3×3	LeakyReLU	$32 \times 512 \times 512$	18k
	Conv 3×3	LeakyReLU	$32 \times 512 \times 512$	9.2k
8	Upsample	-	$32 \times 1024 \times 1024$	-
	Conv 3×3	LeakyReLU	$16 \times 1024 \times 1024$	4.6k
	Conv 3×3	LeakyReLU	$16 \times 1024 \times 1024$	2.3k
	Conv 1×1	sigmoid	$3 \times 1024 \times 1024$	51
				23.1M

Fig. 4



Synthesis of Source Image with Target Image

Input: *source* image S and *target* image T of equal sizes, set of outer facial landmarks $\mathbf{L}$ in image T , standard deviation σ , generated face image resolution r (in our case 1024)

Output: blended *output* image O

$n = \log_2(r)$;

Decompose *source* image S and *target* image T into corresponding Laplacian pyramids $\mathbf{P}(S)_i$ and $\mathbf{P}(T)_i$, where i is a pyramid level, $i \in \langle 1, n \rangle$;

Initialize output pyramid $\mathbf{P}(O)$ for output image O of the same sizes as $\mathbf{P}(T)$ and fill its values with zeroes;

for $i = 1$ **to** n **do**

Compute background mask $\hat{M}_i$ defined as an image of the same size as $\mathbf{P}(T)_i$, where all pixels in the interior of the polygon formed by $\mathbf{L}$ are equal to 0 and 1 otherwise;

$\hat{M}_i = G(\hat{M}_i, \sigma)$, where $G(\hat{M}_i, \sigma)$ denotes gaussian smoothing of $\hat{M}_i$ with standard deviation σ ;

Calculate face mask: $M_i = 1 - \hat{M}_i$;

Copy background from the *target* image to the *output* image:

$\mathbf{P}(O)_i = \mathbf{P}(O)_i + \hat{M}_i \mathbf{P}(T)_i$;

if $i \leq 2$ **then**

Copy face from the *target* image to the *output* image:

$\mathbf{P}(O)_i = \mathbf{P}(O)_i + M_i \mathbf{P}(T)_i$;

else

Copy face from the *source* image to the *output* image:

$\mathbf{P}(O)_i = \mathbf{P}(O)_i + M_i \mathbf{P}(S)_i$;

end

Reconstruct and return output image O from $\mathbf{P}(O)$;

end

Fig. 5