FineGAN : [Unsupervised fine-grained object category discovery]

Purpose: model fine grained object categories by hierarchically disentangling the background, shape & appearance. (No manual annotations) [Fine case]

- Prior work on unsupervised category discovery is not fini-grained & supervised methods are used for fine grained discovery.
- Unsupervised learning enables the discovery of latent structure which may not be done by annotators. [Exploit hierarchy]
- Disentangles the background, object shape & appearance.

* Use information theory to associate each feature to a latent code & then use that condition the rel b/w them in a specific way ⇒ induces hierarchy.
  - BG + Parent image + Child image where the parent & child capture 1 factor of variation each. [capturing factors of variation - info theory]

- By imposing constraints on the rel b/w the parent & child codes, the parent learns the shape & the child learns ⇒ texture
- FineGAN automatically generates masks to help condition the latent codes to only focus on object factors & stitch them together. [Benchmark : CUB, Stanford-dogs, Stanford-cars]
  ↳ Beats SOTA unsupervised techniques [JULE, DEPICT]

- InfoGAN maximises mutual information whereas FineGAN learns a hierarchical disentanglement.

obj fx^n = $L = \lambda L_b + \beta L_p + \gamma L_c$

latent codes: $z \sim N(0,1)$ $b \sim Cat(k=N_b, p=1/N_b)$ $p \sim Cat(k=N_p, p=1/N_p)$ $c \sim Cat(k=N_c, p=1/N_c)$

categorical codes ↑

Imposed hierarchy constraints: (i) $N_p < N_c$

- These help 'p' to capture shape & 'c' to capture appearance.

(ii) For each parent code, there are a few child codes tied to it.

(iii) Sets the background code to be the same as the child code

1. Background Stage

↳ G_b, D_b, D_{aux} ↳ To generate bg, a detector is required that can detect instances of the super-category

↳ The detector is used to ~~detect~~ locate non-object patches in the real image x_i. Train G_b & D_b using:

$L_b = L_{bg\text{-}adv} + L_{bg\text{-}aux}$ → also computed at the patch level

↳ predicts N×N grids with real/fake scores

$$L_{bg\text{-}adv} = \min_{G_b} \max_{D_d} E_x[\log(D_b(x))] + E_{z_b}[\log(1 - D_b(G_b(z, b)))]$$

$$L_{bg\text{-}aux} = \min_{G_b} E_{z,b}[\log(1 - D_{aux}(G_b(z, b)))] \Rightarrow$$ assigns bg & fg probability to each patch

2. Parent Stage: Generate fg entity & stitch it to the bg.

$G_{p,f}$ & $G_{f,M}$ transform F_p into P_f & P_M (foreground, mask)

$$\boxed{P = P_{f,M} + B_M} \Rightarrow \text{Stitching}$$

$P_{f,M} = P_m \odot P_f$ and $B_m = (1 - P_m) \odot B$

Masked foreground — Inverse masked background

- Maximise mutual info $I(p, P_{f,M})$ with D_p approximating $P(p|P_{f,M})$

$$L_p = L_{p\text{-}info} = \max_{D_p, G_{p,f}, G_{p,M}} E_{z,p}[\log(D_p(p|P_{f,M}))]$$

↳ This ensures the decision is based on fg only

- $P_{f,M}$ must not capture any fine grained info and is also common to multiple fine grained categories.

3. Child Stage:

Child code 'c' encodes color/texture and is conditioned on 'p'. 'c' & F_p are fed to G_c. The generated feature rep F_c is used by generators $G_{c,M}$ & $G_{c,f}$ to get C_M & C_f respectively.

$$\boxed{C = C_{f,M} + P_{c,M}} \Rightarrow \text{Stitching}$$

↳ $C_{f,M} = C_m \odot C_f$; $P_{c,m} = (1 - C_M)P$

D_{adv}: Diff b/w real x & gen C

D_c: Approximate $P(c|C_{f,M})$

$L_c = L_{adv} + L_{info}$

$L_{adv} = E_x \log(D_{adv}(x)) + E_{z,p,c}[\log(1 - D_{adv}(C))]$

$L_{info} = \max_{D_c, G_{c,f}, G_{c,M}} E_{z,p,c}[\log D_c(c|C_{f,M})]$