* Ziel
  + Inpainting
* Einführung
  + Bestehende Inpainting Algorithmen sind auf spezifische Masks trainiert
  + Training mit pixel-wise und perceptual losses führen häufig zu simplen Füllungen
* Model
  + Sie haben kein mask-conditional model, sondern sie samplen nur diese spezifischen pixels im reverse prozess
    - Ein intermediate step setzen sie mit zwei Teilen zusammen
    - 1. Teil sind bekannte Pixels ausserhalb der Maske: Dort fügen sie dem originalen Bild einfach die Noise für diesen Step hinzu
    - 2. Teil sind die unbekannten Pixels innerhalb der Maske: Für diese fügen sie ganzes Bild vom vorherigen Step in das model ein und extrahieren vom Output nur die Pixels innerhalb der Maske.
    - Anschliessend setzen sie die beiden Teile zusammen und haben so Bild vom nächsten Step
    - Ein Bild, das Text, Schrift, Screenshot, weiß enthält.

      Automatisch generierte Beschreibung
  + Vorteile
    - Generalize Network zu jeder Maske während Inference -> ich glaube bei inpainting-conditioned network trainiere ich auf spezifische Masken
    - Erlaubt Netzwerk mehr semantic generation capabilities zu lernen, weil es starke image synthesis prior hat.
  + Problem: Inpainted Region hat zwar texture vom darumliegenden Bild, jedoch ist es semantisch inkorrekt
  + Ein Bild, das Haustier, Hunderasse, Shetland Sheepdog, Säugetier enthält.

    Automatisch generierte Beschreibung
  + Erklärung
    - Wenn der 1. Teil generiert wird d.h. den bekannten Pixels noise hinzugefügt. Dann wird dies ohne Kenntnis von den generierten Pixel für den mask bereich gemacht
    - Model versucht zwar bei jedem Step diese disharmony wieder auszugleichen, jedoch übenehmen wir immer nur den mask Bereich d.h. es kann nie richtig convergen. Insbesondere sind Changes durch noise schedule restricted.
    - Aus diesem Grund braucht model mehr Zeit um dishamrnoy auszugleichen innerhalb eines denoising steps bevor es zum nächsten step schreitet
  + Lösung: Resampling
    - Wir generieren x\_(t-1) wie vorher beschrieben, aber diffusen dann nochmals indem ich noise wieder hinzufüge (wie ich es im forward process in diesem Schritt machen würde)
    - Jeh mehr resampling steps desto bessere Qualität. Gut ist ab ca. 10 resampling steps
    - Resampling steps definieren wie viel ich nach hinten und nach vorne gehe, bevor ich weiter gehe (hinten/vorne -> rauf/runter im Bild)
    - Jump lengths definieren wie viel forward steps ich mache, bevor ich wieder reverse steps mache (forward in Bezug auf Noise hinzufügen)
* Ein Bild, das Text, Screenshot, Reihe, Diagramm enthält.

  Automatisch generierte Beschreibung
  + - Resampling steps sind Anzahl Zick Zack bevor es einen Schritt nach unten geht
    - Jump lengths definiert wie lang die Zick Zacks sind
* Training
  + 256x256 Crops auf 4xV100 GPU
  + Sie nutzten T=250 timesteps mit r=10 resmapling mit jumpy size j=10
* Evaluation
  + Human evaluation
    - Sie zeigten Bilder von zwei Inpainting Algorithms und man musste sagen, welche Variante besser ist
  + Perceptual metric LPIPS
    - Distance metric based on deep feature space of AlexNet
  + Diversity Score
  + Weitere Evaluation anhand LaMa settings für Wide und Narrow masks
    - Wide masks für grosse Bereiche zum Inpainten e.g. 50%
      * Viele andere models struggeln hier
    - Und narrow masks zum schauen ob es feine Strukturen füllen kann
      * Andere modelle waren blurry
  + Ein Bild, das Collage, Screenshot, Fotomontage enthält.

    Automatisch generierte Beschreibung