* Ziel
  + Bilder vergleichen und anhand einer Metric bestimmen welche Bilder ähnlich sind
  + Es gibt bereits Metriken, aber wir möchten eine, welche besser mit Menschen übereinstimmt
* Bestehende Metriken
  + PSNR und SSIM
  + Diese vergleichen u.a. Bilder pixelweis (deshalb gelten blurred images als ähnlich)
* Menschliche Einschätzung von Ähnlichkeit basiert auf
  + High-order image structure
  + Context-dependent
  + stellen möglicherweise nicht wirklich eine Entfernungsmetrik dar
* Ist ein Bild verschwummen dann ist l2 distance klein, aber visuell ist Unterschied sehr gross
* Als metric können auch deep features von DL Netzwerken genommen werden (e.g. VGG) und dann die l2 distance von diesen Features als perceptual loss
  + Man hat zwei Bilder und lässt diese durch ein Netzwerk
  + Der resultierende Feature stack von beiden Bildern wird entlang der channel Richtung normalisiert und dann subtrahiert
  + Anschliessend l2 norm nehmen und dann über spatial dimension normalisieren (l2 pro neuron berechnen und averagen)
  + Dann am Schluss nochmals über alle Layer averagen
  + Ein Bild, das Diagramm, Reihe, Screenshot, Text enthält.

    Automatisch generierte Beschreibung
* Dataset
  + Human judgement mit zwei Tests
    - Two alternative forced choice (2AFC): Welche von zwei Distortions/Bilder sind ähnlicher zu einer Referenz
    - Just noticeable difference (JND): Sind zwei Patches gleich oder unterschiedlich
  + Distortions
    - Traditionelle Distortions
      * Photometric
        + lightness shift, color shift, contrast, saturation
      * Noise
        + uniform white noise, Gaussian white, pink, & blue noise, Gaussian colored (between violet and brown) noise, checkerboard artifact
      * Blur
        + Gaussian, bilateral filtering
      * Spatial
        + shifting, affine warp, homography, linear warping, cubic warping, ghosting, chromatic aberration,
      * Compression
        + Jpeg
    - CNN based
      * Dies sind trainierte models mit gewissen Tasks e.g. autoencoding, denoising, colorization und superresolution. Diese Art von Artifacts sind wichtig, weil sie typische deep learning model artifacts sind.
      * Hier werden vorallem mit Netzwerk Parameter gespielt
      * Input corruption
        + null, pink noise, white noise, color removal, downsampling
      * Generator network architecture
        + # layers, # skip connections, # layers with dropout, force skip connection at highest layer, upsampling method, normalization method, first layer stride # channels in 1st layer, max # channels
      * Discriminator
        + number of layers
      * Loss/Learning
        + weighting on oixel-wise (`1), VGG, discriminator losses, learning rate
  + Distorted image patches from real algorithms
    - (Keine direkte distortions generieren, sondern bestehende Algorithmen nutzen, welche indirekt distortions erzeugen)
    - Superresolution algorithms
    - Frame interpolation
    - Video deblurring
    - Colorization
* Netzwerke
  + Sie verglichen deep features von verschiedenen Netzwerke und nutzten l2 distance von diesen features als metric
  + Netzwerke/Metrics
    - Alte metrics
    - Netzwerk mit random weights
    - Self-supervised trainierte Netzwerke
    - Supervised trainierte Netzwerke
      * Netzwerke
        + Squeeze
        + Alex
        + VGG
      * Variationen
        + Original weights
        + Additional linear weighting w trained on their dataset

Diese linear weights gewichten die channels und ist nicht ein Layer am Ende des Netzwerks

Weights überall =1 gibt originale Netzwerke

Ein Bild, das Diagramm, Reihe, Screenshot, Plan enthält.

Automatisch generierte Beschreibung

* + - * + Finetuning netzwerk on their dataset
        + Training from scratch on their dataset
* Conclusion
  + Alle Netzwerke übertrumpfen alte Metrics
  + Ein gutes Feature für einen Task e.g. semantic task ist auch ein gutes Feature für einen anderen Task e.g. self-supervised
  + Die finetuned Netzwerke sind am besten für ihre distrotions, jedoch verglichen mit real world algorithm die linear weighting gibt die besten Resultate
* Background
  + Die Leute vom Research Paper wollten ein colourizing task machen d.h. schwarz-weiss Bilder einfärben
    - Jedoch bekamen sie schlechte Resultate mit L2 Regression
      * Netzwerk kann schätzen welche Farben wahrscheinlich sind und L2 regression nimmt einfach Average davon
      * Bessere Variante ist mit Classification d.h. colour space wird diskretitiert und für jeden Pixel wird pdf erstellt über colour space. Anschliessend wird die wahrscheinlichste Farbe genommen.
      * -> d.h. loss function hat grossen Einfluss
      * Ein Bild, das draußen, Vogel, Text, sitzen enthält.

        Automatisch generierte Beschreibung
    - Netzwerk hat komische Fehler gemacht
      * Ein Bild, das Hund, Säugetier, Hunderasse, Haustier enthält.

        Automatisch generierte Beschreibung
      * Das Netzwerk hat den Mund vom Hund komisch pink eingefärbt
      * Dies kommt davon weil bei fast allen Trainingdaten der Mund immer offen war. D.h. das Netzwerk hat nicht nur Einfärbung gelernt anhand schwarz-weiss Fotos, sondern auch Konzept eines Hundekopfs und dessen Farbverteilung.
  + Deep Netzwerk sind nicht shift invariance
    - Features sind anscheinend unterschiedlich zwischen zwei gleichen Bilder, wenn diese ein wenig verschoben sind
    - Dies ist überraschend weil eigentlich gilt
      * Convolutions sind shift-equivariant
      * Pooling sind shift-invariant
    - Problem: striding
      * Weil ich andere samples nehmen
      * Problematisch für max-pooling
    - Lösung: blurPool (Antialiasing)
      * Definieren wo das Maximum von pixel ist, dann pixel blurren und dann der Maximum Index von vorher nehmen
      * -> macht problem nicht mehr so schlimm
* Fragen
  + Wie können random initialisierte netzwerke irgend eine aussagekraft geben über feature distance? Oder generieren auch random convolutional netzwerke gewisse features?