Exercise 2 of Medical Image Processing SS2014

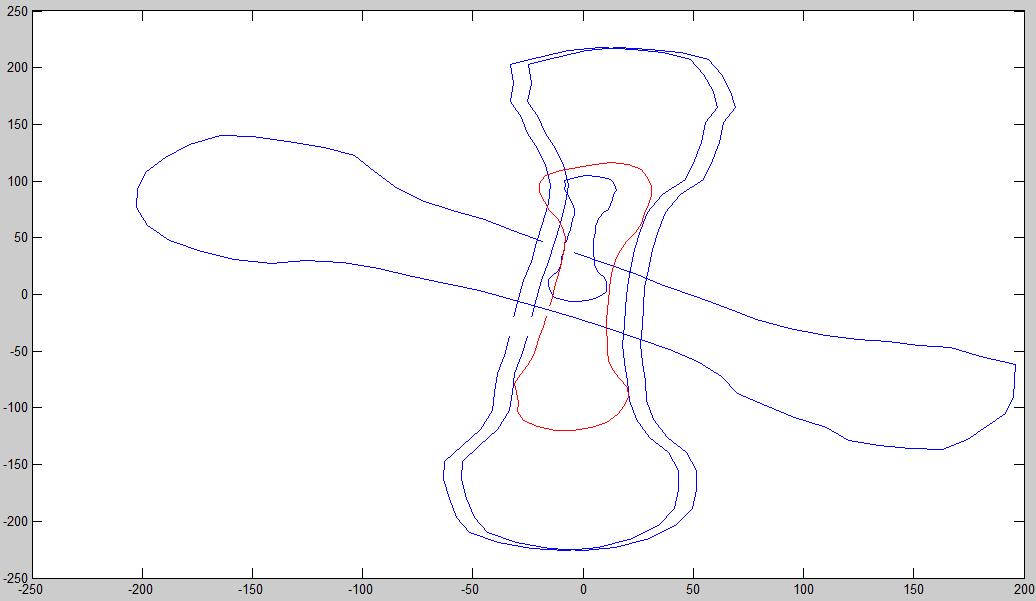
By: Dr. René Donner

Team: Stephan Spindler 0828742, S. Ehsan Hosseini 1028660

1. **Shape Model:**

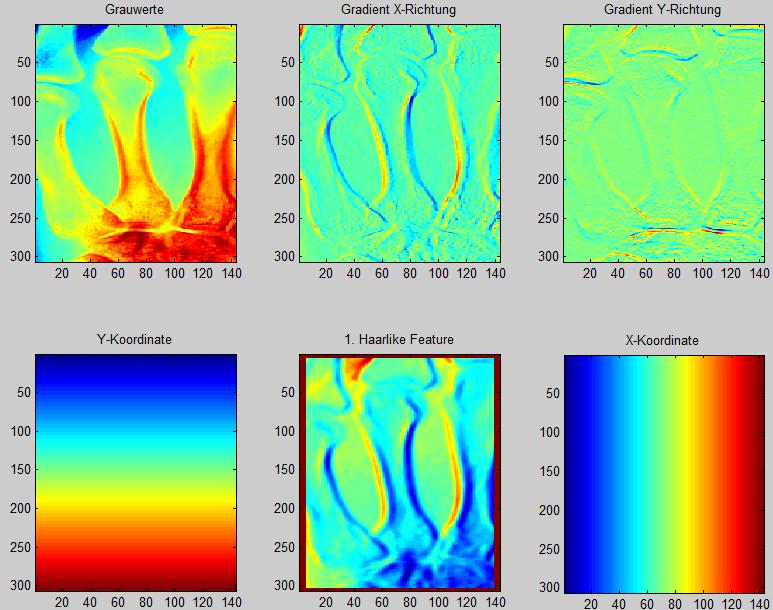
* **generateShape.m**
* **Plotten Sie analog zum ersten Beispiel Shapes für mehrere Werte von Skalierung und Rotation:**

Reference Bone: **Rot**

****

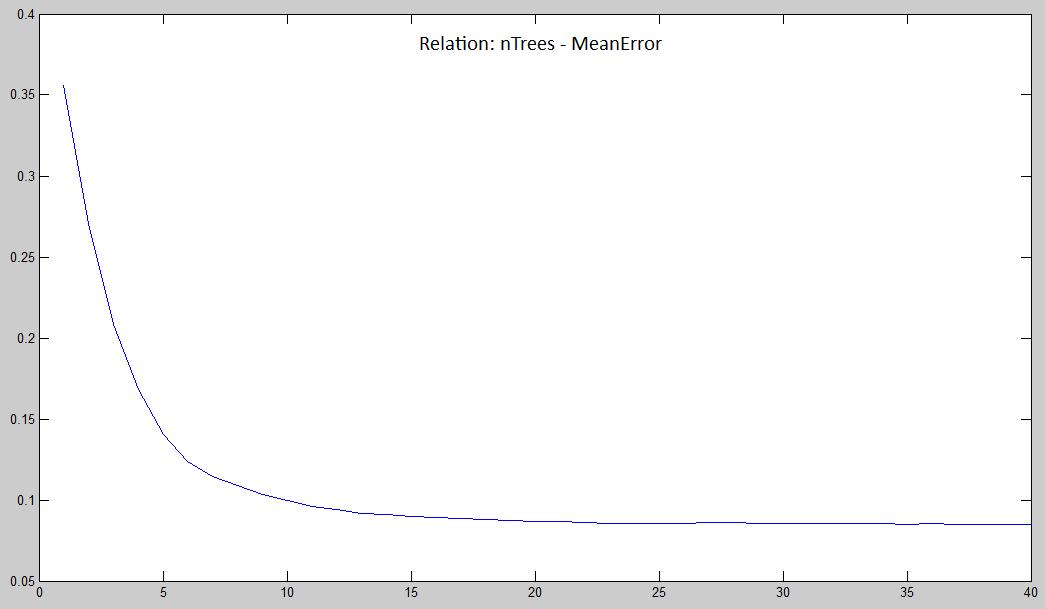
1. **Featureberechnung:**

* **ComputeFeatures.m**

****

1. **Klassifikation & Feature-Selection:**
2. **train.m**
3. **Untersuchen und Interpretieren Sie den Einuss der Anzahl von Trees mittels oobError**

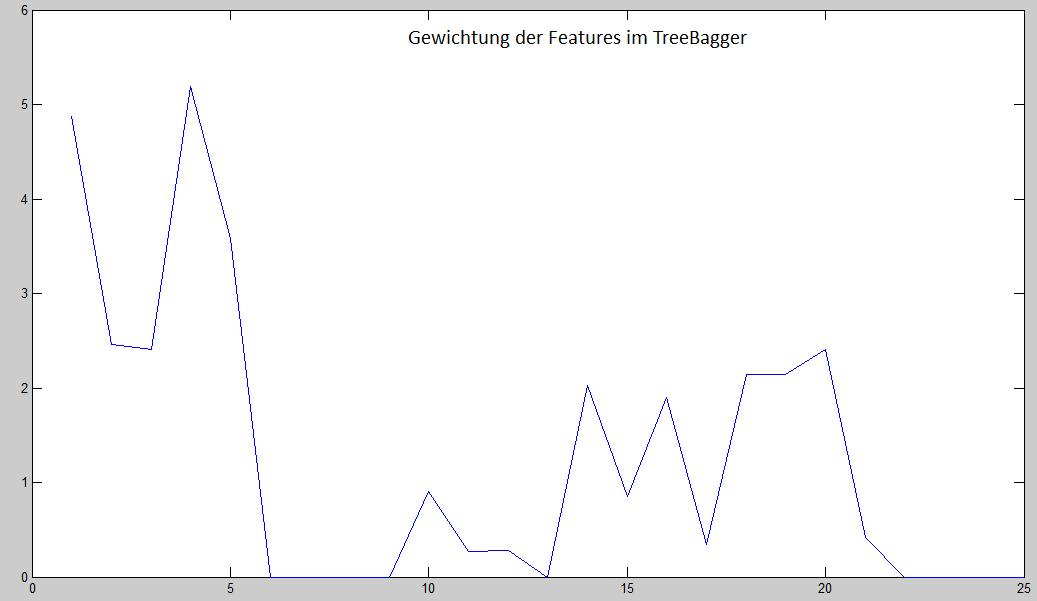
* Der nachfolgende Plot zeigt die Relation zwischen der Anzahl der verwendeten Trees im **TreeBagger** zum resultierenden MeanError. Ab n Trees~ 20 konvergiert der Error gegen 0.015. Das heißt, dass eine Anzahl n Trees>20 keine größeren Auswirkungen auf die Fehlerabweichungen haben.

****

1. **Untersuchen und Interpretieren Sie die Wichtigkeit der verschiedenen Features mittels plot(rf.OOBPermutedVarDeltaError).**

* Der Plot zeigt die Relation zwischen verwendeten Features und jeweiliger Gewichtung im TreeBagger. Die Features Gradientenstärke, Haar-like Feature 6 (der Grauwerte), erstes und fünftes Haar-like Feature der Gradientenstärke und die x- und y- Koordinaten haben die größte Gewichtung.

1. Gradientenstärke (im Bild: Anstiegsstärke der Intensität)
2. Haar-like Feature (möglicher Grund für hohe Gewichtung des einzelnen Haar-Features: Hohe Anzahl gefundener Kanten, Texturänderung und Hell-Dunkel- Regionen)
3. - x- und y-Koordinaten (Lokalisierung)



1. **Shape Particle Filters:**
2. siehe **train.m** & **myPredict.m**
3. &
4. (Closure) siehe **myOptimize.m**
5. Vier Steps:

1. Initialisation: Grenzbestimmung der Parameter.

2. Mutation: Expandierung auf weitere Zufallsvektoren mit ungleichen Indizes.

Distanzen werden aufaddiert (Gebervektor).

3. Rekombination: Erstellung des Trial-Vektors aus Zielvektor und aufaddiertem

Gebervektor.

4. Selektion: Vergleich Zielvektor mit Trial-Vektor; niedrigste Abweichung wird

weitergegeben -> erneute Mutation bis zur Konvergenz.

Die Funktion optimize.m definiert in der Zeile 17-19 die Variante der Differential Evolution (nach DE/rand/1/bin).Das heißt, ein zufällig ausgewählter Gebervektor wählt ein Lösungspaar (binominale Rekombination).

Umso weniger Evaluierungen verwendet werden, desto schneller konvergiert der Algorithmus, mit abnehmender Genauigkeit.