CNRS - TELECOM ParisTech at ImageCLEF 2013 Scalable Concept Image Annotation Task

Hichem SAHBI

CNRS TELECOM ParisTech, Paris

ImageCLEF 2013

Sep 25th 2013

Photo Annotation in Computer Vision

- More progress needs to be done by seeking :
  - Better visual features (physical aspects of scenes, signal processing, etc.)
  - Better learning algorithms (sample size problem, sparse solutions, etc.)
  - Extra knowledges and priors (context, ontologies, 3D information, etc.)

Photo Annotation in Computer Vision

- More progress needs to be done by seeking :
  - Better visual features (physical aspects of scenes, signal processing, etc.)
  - Better learning algorithms (sample size problem, sparse solutions, etc.)
  - Extra knowledges and priors (context, ontologies, 3D information, etc.)

Photo Annotation in Computer Vision

- More progress needs to be done by seeking :
  - Better visual features (physical aspects of scenes, signal processing, etc.)
  - Better learning algorithms (sample size problem, sparse solutions, etc.)
  - Extra knowledges and priors (context, ontologies, 3D information, etc.)

Photo Annotation in Computer Vision

- More progress needs to be done by seeking :
  - Better visual features (physical aspects of scenes, signal processing, etc.)
  - Better learning algorithms (sample size problem, sparse solutions, etc.)
  - Extra knowledges and priors (context, ontologies, 3D information, etc.)

Context (Analogy with Speech)

- Large corpus in speech recognition and machine translation are available for training but one cannot successfully recognize/translate single words.
- Context information (language models, context-dependent MFCC feature modeling, context-dependent phrase translation) has played an important role in improving the results.
- Need to learn the appearance of primitives in computer vision (interest points, objects, etc.) not only from many instances but also from different contexts.

Context (Analogy with Speech)

- Large corpus in speech recognition and machine translation are available for training but one cannot successfully recognize/translate single words.
- Context information (language models, context-dependent MFCC feature modeling, context-dependent phrase translation) has played an important role in improving the results.
- Need to learn the appearance of primitives in computer vision (interest points, objects, etc.) not only from many instances but also from different contexts.

Context (Analogy with Speech)

- Large corpus in speech recognition and machine translation are available for training but one cannot successfully recognize/translate single words.
- Context information (language models, context-dependent MFCC feature modeling, context-dependent phrase translation) has played an important role in improving the results.
- Need to learn the appearance of primitives in computer vision (interest points, objects, etc.) not only from many instances but also from different contexts.

Outline

- Introduction
- Kernels (on non vectorial data)
- Context in kernel design
- Context dependent image annotation
- Conclusion and take home message

Outline

- Introduction
- Kernels (on non vectorial data)
- Context in kernel design
- Context dependent image annotation
- Conclusion and take home message

Outline

- Introduction
- Kernels (on non vectorial data)
- Context in kernel design
- Context dependent image annotation
- Conclusion and take home message

About Kernels

- Kernels are symmetric, continuous and positive (semi) definite similarity functions widely used in Machine Learning (SVMs, KPCA, SVR, etc.).
- Define $\mathcal{X}$ as an input space and $X, X' \in \mathcal{X}$. $k : \mathcal{X} \times \mathcal{X} \longrightarrow \mathbb{R}^+$. Here $k(X, X')$ is referred to as similarity between X and X'.
- Classic kernels (on vectorial data) : linear $k(X, X') = \langle X, X' \rangle$, polynomial $k(X, X') = (1 + \langle X, X' \rangle)^p$, Gaussian, etc.
- if k is p.s.d, then $\exists \phi : \mathcal{X} \rightarrow \mathcal{H}$, $k(X, X') = \langle \phi(X), \phi(X') \rangle$.
- Kernels can be designed using closure operations (additions, products, exponentiation, etc.)

About Kernels

- Kernels are symmetric, continuous and positive (semi) definite similarity functions widely used in Machine Learning (SVMs, KPCA, SVR, etc.).
- Define $\mathcal{X}$ as an input space and $X, X' \in \mathcal{X}$.
$k : \mathcal{X} \times \mathcal{X} \longrightarrow \mathbb{R}^+$. Here $k(X, X')$ is referred to as similarity between X and X'.
- Classic kernels (on vectorial data) : linear $k(X, X') = \langle X, X' \rangle$, polynomial $k(X, X') = (1 + \langle X, X' \rangle)^p$, Gaussian, etc.
- if k is p.s.d, then $\exists \phi : \mathcal{X} \rightarrow \mathcal{H}$, $k(X, X') = \langle \phi(X), \phi(X') \rangle$.
- Kernels can be designed using closure operations (additions, products, exponentiation, etc.)

About Kernels

- Kernels are symmetric, continuous and positive (semi) definite similarity functions widely used in Machine Learning (SVMs, KPCA, SVR, etc.).
- Define $\mathcal{X}$ as an input space and $X, X' \in \mathcal{X}$. $k : \mathcal{X} \times \mathcal{X} \longrightarrow \mathbb{R}^+$. Here $k(X, X')$ is referred to as similarity between X and X'.
- Classic kernels (on vectorial data) : linear $k(X, X') = \langle X, X' \rangle$, polynomial $k(X, X') = (1 + \langle X, X' \rangle)^p$, Gaussian, etc.
- if k is p.s.d, then $\exists \phi : \mathcal{X} \rightarrow \mathcal{H}$, $k(X, X') = \langle \phi(X), \phi(X') \rangle$.
- Kernels can be designed using closure operations (additions, products, exponentiation, etc.)

About Kernels

- Kernels are symmetric, continuous and positive (semi) definite similarity functions widely used in Machine Learning (SVMs, KPCA, SVR, etc.).
- Define $\mathcal{X}$ as an input space and $X, X' \in \mathcal{X}$. $k : \mathcal{X} \times \mathcal{X} \longrightarrow \mathbb{R}^+$. Here $k(X, X')$ is referred to as similarity between X and X'.
- Classic kernels (on vectorial data) : linear $k(X, X') = \langle X, X' \rangle$, polynomial $k(X, X') = (1 + \langle X, X' \rangle)^p$, Gaussian, etc.
- if k is p.s.d, then $\exists \phi : \mathcal{X} \rightarrow \mathcal{H}$, $k(X, X') = \langle \phi(X), \phi(X') \rangle$.
- Kernels can be designed using closure operations (additions, products, exponentiation, etc.)

About Kernels

- Kernels are symmetric, continuous and positive (semi) definite similarity functions widely used in Machine Learning (SVMs, KPCA, SVR, etc.).
- Define $\mathcal{X}$ as an input space and $X, X' \in \mathcal{X}$. $k : \mathcal{X} \times \mathcal{X} \longrightarrow \mathbb{R}^+$. Here $k(X, X')$ is referred to as similarity between X and X'.
- Classic kernels (on vectorial data) : linear $k(X, X') = \langle X, X' \rangle$, polynomial $k(X, X') = (1 + \langle X, X' \rangle)^p$, Gaussian, etc.
- if k is p.s.d, then $\exists \phi : \mathcal{X} \to \mathcal{H}$, $k(X, X') = \langle \phi(X), \phi(X') \rangle$.
- Kernels can be designed using closure operations (additions, products, exponentiation, etc.)

Kernels for non-Vectorial Data (Graphs, Trees, Strings, etc.)

- Order and length insensitive kernels
  - Kullback Leibler divergence kernels (Kondor & Jebara, 2003).
  - Battacharyya affinity (Moreno et al. 2003).
  - Principal angles (Wolf & Shashua, 2003).
  - Subset kernels (Shashua & Hazan, 2004).
- Alignment kernels
  - Max kernel (Wallraven et al. 2003).
  - Circular shift kernel (Lyu 2005).
  - Intermediate kernel (Boughorbel 2005).
  - Pyramid match kernel (Grauman & Trevor 2007).
  - Dynamic programming kernel (Bahlmann et al. 2002).

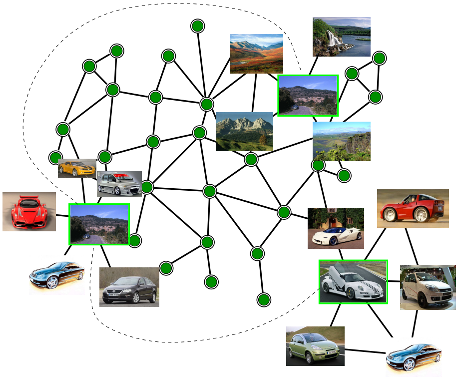
Outline

- Introduction
- Kernels (on non vectorial data)
- Context in kernel design
- Context dependent image annotation
- Conclusion and take home message

Outline

- Introduction
- Kernels (on non vectorial data)
- Context in kernel design
- Context dependent image annotation
- Conclusion and take home message

Context-Free Kernel Deficiency



[Sahbi & Li, ACCV 10, Sahbi & al., PAMI 2011]

Context Dependent Kernels (Formulation)

- An image database $\mathcal{X} = \{x_1, \dots, x_n\}$ is modeled as a graph, nodes are pictures and edges are typed (tagged) links from an alphabet Ω.

$$\min_k \sum_{x_i, x_j} k(x_i, x_j)\, d(x_i, x_j) + \frac{\beta}{2} \sum_{x_i, x_j} k(x_i, x_j)^2$$

$$+ \quad \alpha \sum_{x_i, x_j} k(x_i, x_j) \left(- \sum_{c \in \Omega} \sum_{\substack{x_k \in \mathcal{N}^c(x_i), \\ x_\ell \in \mathcal{N}^c(x_j)}} k(x_k, x_\ell) \right)$$

- $\min_{\mathbf{K}} \quad tr(-\mathbf{K}\mathbf{S}') - \alpha \sum_c tr(\mathbf{K}\mathbf{P}_c\mathbf{K}'\mathbf{P}'_c) + \frac{\beta}{2}\|\mathbf{K}\|^2,$

$\mathbf{K}_{i,j} = k(x_i, x_j)$, $\mathbf{P}_{c,i,k} = \mathbb{1}_{\{x_k \in \mathcal{N}^c(x_i)\}}$.

Context Dependent Kernels (Formulation)

- An image database $\mathcal{X} = \{x_1, \dots, x_n\}$ is modeled as a graph, nodes are pictures and edges are typed (tagged) links from an alphabet Ω.

$$\min_k \sum_{x_i, x_j} k(x_i, x_j)\, d(x_i, x_j) + \frac{\beta}{2} \sum_{x_i, x_j} k(x_i, x_j)^2$$
$$+ \quad \alpha \sum_{x_i, x_j} k(x_i, x_j) \left(- \sum_{c \in \Omega} \sum_{\substack{x_k \in \mathcal{N}^c(x_i), \\ x_\ell \in \mathcal{N}^c(x_j)}} k(x_k, x_\ell) \right)$$

- $\min_{\mathbf{K}} \quad tr(-\mathbf{K}\mathbf{S}') - \alpha \sum_c tr(\mathbf{K}\mathbf{P}_c\mathbf{K}'\mathbf{P}'_c) + \frac{\beta}{2} \|\mathbf{K}\|^2,$
$\mathbf{K}_{i,j} = k(x_i, x_j)$, $\mathbf{P}_{c,i,k} = \mathbb{1}_{\{x_k \in \mathcal{N}^c(x_i)\}}$.

Kernel Design (Solution)

Proposition (1)

The optimization problem admits a solution $\tilde{\mathbf{K}}$, which is the limit of

$$\mathbf{K}^{(t)} = \mathbf{K}^{(0)} + \gamma \sum_c \mathbf{P}_c \mathbf{K}^{(t-1)} \mathbf{P}'_c, \qquad \textit{(Entry-wise.)}$$

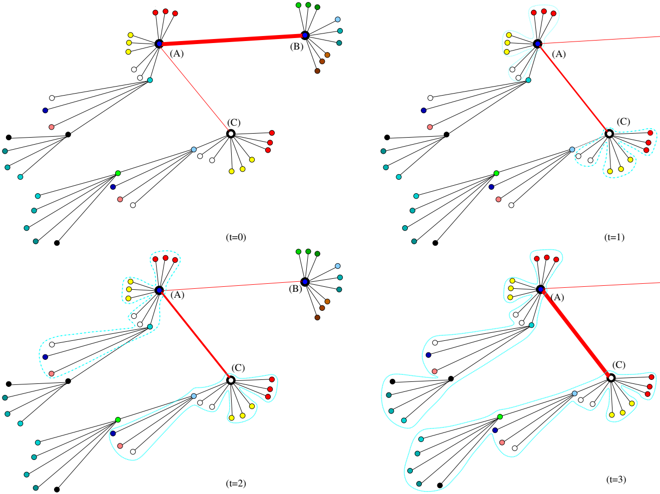
Provided that the following inequality holds, convergence is guaranteed

$$\gamma < \| \sum_c \mathbf{P}_c \mathbf{1}_{mm} \mathbf{P}'_c \|_1^{-1}, \tag{1}$$

$$\textit{with} \quad \gamma = \frac{\beta}{\alpha} \tag{2}$$

Proof in [Sahbi, ICVS 2013].

Influence of Context : Example



Explicit Positive Definiteness

Proposition (2)

The similarity functions associated to $\mathsf{K}^{(t+1)}$, $(t = 0, 1, \dots)$ *defined, in proposition earlier, as*

$$\mathsf{K}^{(t+1)} = \big(\mathsf{K}^{(0)} + \gamma \sum_c \mathsf{P}_c \; \mathsf{K}^{(t)} \; \mathsf{P}'_c\big)$$

are explicit p.s.d kernels.

Explicit Positive Definiteness (proof.)

Proof of proposition 2.

by induction ; $\mathbf{K}^{(0)}_{x,x'} = (\Phi'^{(0)}\Phi^{(0)})_{x,x'}$ is explicit p.s.d as it is per definition p.s.d and the mapping $\Phi^{(0)}$ are known and finite dimensional.
Assuming $\mathbf{K}^{(t)}_{x,x'}$ explicit p.s.d, we obtain

$$\begin{aligned}\mathbf{K}^{(t+1)}_{x,x'} &= \Big(\Phi'^{(0)}\Phi^{(0)} + \gamma \sum_c \mathbf{P}_c \mathbf{K}^{(t)} \mathbf{P}'_c\Big)_{x,x'} \\ &= \Big(\Phi'^{(0)}_p\Phi^{(0)}_q + \gamma \sum_c \mathbf{P}_c \Phi'^{(t)}\Phi^{(t)} \mathbf{P}'_c\Big)_{x,x'} \\ &= \big(\Phi'^{(t+1)}\Phi^{(t+1)}\big)_{x,x'}\end{aligned}$$

$$\text{with} \quad \Phi^{(t+1)} = \begin{pmatrix} \Phi'^{(0)} & \gamma^{\frac{1}{2}}\,\mathbf{P}_1\Phi'^{(t)} & \dots & \gamma^{\frac{1}{2}}\mathbf{P}_{|\Omega|}\Phi'^{(t)} \end{pmatrix}'.$$

Since $\Phi^{(t)}$ is f.d, $\Phi^{(t+1)}$ is also f.d so $\mathbf{K}^{(t+1)}_{x,x'}$ is explicit p.s.d. □

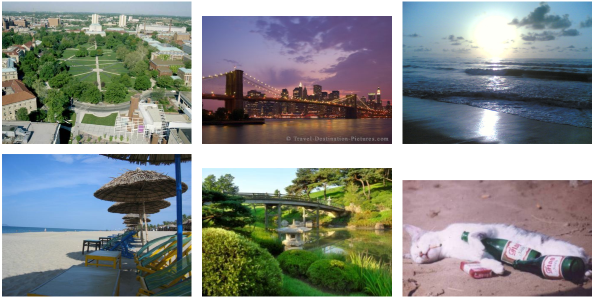
Outline

- Introduction
- Kernels (on non vectorial data)
- Context in kernel design
- Context dependent image annotation
- Conclusion and take home message

Outline

- Introduction
- Kernels (on non vectorial data)
- Context in kernel design
- Context dependent image annotation
- Conclusion and take home message

ImageCLEF 2013



- 250k (unlabeled) training images, 1k images for dev and 2k images for test. Nbr of concepts : 116.
- Nbr of submitted runs : 6 based on kernel SVMs (2 without context and 4 with context).
- 13 participants, 58 participant runs.

ImageCLEF 2013 : Training data generation

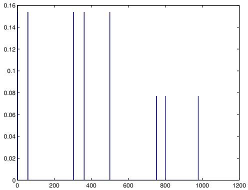
- For a given concept (among the 116 concepts), we extract a training set, by collecting among the 250k images those which include that concept, in their meta-data (webupv13_train_textual.keywords).

  cat -n webupv13_train_textual.keywords | grep aerial | more
  130 ergot 52 g aerially 100 g
  1516 aerials 1 g
  1569 aerials 2 g
  2363 london 72 g aerial 107 b
  2545 aerials 60 b spun 151 b

- We applied some very simple morphological expansions in order to increase the recall : baby -> babies, galaxy -> galaxies, etc.

ImageCLEF 2013 : Context

- **Context matrix generation :** we design a left stochastic adjacency matrix (denoted $\mathbf{P} = \mathbf{D}^{-1}\mathbf{C}$) between images with each entry proportional to the number of shared keywords in the meta-data of the underlying images (using files devel_iurls.txt, devel_rurls.txt, train_iurls.txt, train_rurls.txt, etc.)



- We use this adjacency matrix in order to build our context dependent kernels.

ImageCLEF 2013 : Submitted Runs (I)

- All our submitted runs are based on SVMs.
- We used only the visual features provided in this imageCLEF task including GIST, Color Histograms, SIFT, C-SIFT, RGB-SIFT and OPPONENT-SIFT.
- We trained "one-versus-all" SVM classifiers for each concept ω.

$$f_\omega(x) = \frac{1}{N} \sum_{\ell=1}^{N} 1_{\{g_{\ell(x)} \geq 0\}} \in [0,1], \quad (N = 10 \ \text{ in practice})$$

$$\omega \in x \quad \text{iff} \quad f_\omega(x) \geq \tau$$

ImageCLEF 2013 : Submitted Runs (II)

- For all the submitted runs (6 runs), the only difference resides in the used kernels.
- **Run 1.** we linearly combine 7 gram matrices into a single one (not MKL just a combination of kernels with uniform weights). We plug the resulting kernel into SVMs for training and testing.
  $\omega \in x \quad \text{iff} \quad f_\omega(x) \geq \tau$ (with $\tau = 0.5$).
- **Run 2.** the setting of this run is exactly the same as run 1 except that the cut-off threshold τ is set to 1.

ImageCLEF 2013 : Submitted Runs (II)

- For all the submitted runs (6 runs), the only difference resides in the used kernels.
- **Run 1.** we linearly combine 7 gram matrices into a single one (not MKL just a combination of kernels with uniform weights). We plug the resulting kernel into SVMs for training and testing.
  $\omega \in x \quad \text{iff} \quad f_\omega(x) \geq \tau$ (with $\tau = 0.5$).
- **Run 2.** the setting of this run is exactly the same as run 1 except that the cut-off threshold τ is set to 1.

ImageCLEF 2013 : Submitted Runs (II)

- For all the submitted runs (6 runs), the only difference resides in the used kernels.
- **Run 1.** we linearly combine 7 gram matrices into a single one (not MKL just a combination of kernels with uniform weights). We plug the resulting kernel into SVMs for training and testing.
  $\omega \in x \quad \text{iff} \quad f_\omega(x) \geq \tau$ (with $\tau = 0.5$).
- **Run 2.** the setting of this run is exactly the same as run 1 except that the cut-off threshold τ is set to 1.

ImageCLEF 2013 : Submitted Runs (III)

- **Run 3.** we evaluate our CDK defined as $\mathbf{K}^{(t+1)} = \mathbf{K}^{(0)} + \gamma \ \mathbf{P}\mathbf{K}^{(t)}\mathbf{P}'$, with $\gamma \geq 0$ ($\gamma = 1$ in practice). We plug the resulting kernel into SVMs for training and testing.
  $\omega \in x \quad \text{iff} \quad f_\omega(x) \geq \tau$ (with $\tau = 0.5$).
- **Run 4.** the setting of this run is exactly the same as run 3 except that the cut-off threshold τ is set to 1.

ImageCLEF 2013 : Submitted Runs (III)

- **Run 3.** we evaluate our CDK defined as $\mathbf{K}^{(t+1)} = \mathbf{K}^{(0)} + \gamma \ \mathbf{P}\mathbf{K}^{(t)}\mathbf{P}'$, with $\gamma \geq 0$ ($\gamma = 1$ in practice). We plug the resulting kernel into SVMs for training and testing.
  $\omega \in x \quad \text{iff} \quad f_\omega(x) \geq \tau$ (with $\tau = 0.5$).

- **Run 4.** the setting of this run is exactly the same as run 3 except that the cut-off threshold τ is set to 1.

ImageCLEF 2013 : Submitted Runs (IV)

- **Run 5.**
  - we first evaluate CDK $\mathbf{K}^{(t+1)} = \mathbf{K}^{(0)} + \gamma\ \mathbf{P}\mathbf{K}^{(t)}\mathbf{P}'$ and explicit CDK maps $\Phi^{(t+1)} = \left(\Phi'^{(0)} \quad \gamma^{\frac{1}{2}}\ \mathbf{P}\Phi'^{(t)}\right)'$ for every visual feature.
  -we compute histogram intersection kernel on these CDK maps.
  -we linearly combine the resulting kernels with uniform weights.
  We plug the resulting kernel into SVMs for training and testing.
  $\omega \in x \quad \text{iff} \quad f_\omega(x) \geq \tau$ (with $\tau = 0.5$).

- **Run 6.** the setting of this run is exactly the same as run 5 except that the cut-off threshold τ is set to 1.

ImageCLEF 2013 : Submitted Runs (IV)

- **Run 5.**
  - we first evaluate CDK $\mathbf{K}^{(t+1)} = \mathbf{K}^{(0)} + \gamma\ \mathbf{P}\mathbf{K}^{(t)}\mathbf{P}'$ and explicit CDK maps $\Phi^{(t+1)} = \left(\Phi'^{(0)} \quad \gamma^{\frac{1}{2}}\ \mathbf{P}\Phi'^{(t)}\right)'$ for every visual feature.
  -we compute histogram intersection kernel on these CDK maps.
  -we linearly combine the resulting kernels with uniform weights.
  We plug the resulting kernel into SVMs for training and testing.
  $\omega \in x \quad \text{iff} \quad f_\omega(x) \geq \tau$ (with $\tau = 0.5$).

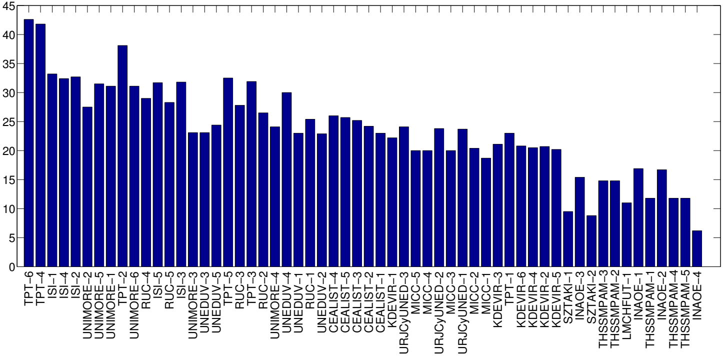
- **Run 6.** the setting of this run is exactly the same as run 5 except that the cut-off threshold τ is set to 1.

Performance and Rank : (ImageCLEF 2013)

Participants (**TPT** : CNRS - Telecom ParisTech, Paris, France.)

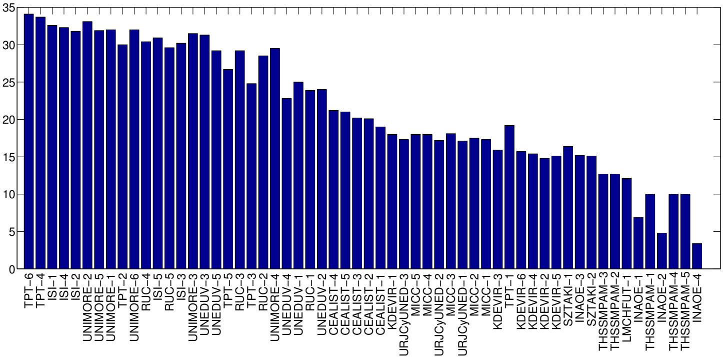
- **ISI** : Tokyo U., Japan.
- **UNIMORE** : U. of Modena and Reggio Emilia, Italy.
- **RUC** : Renmin U. of China.
- **UNEDUV** : National U. of Distance Education at Spain.
- **CEALIST** : CEA, France.
- **KDEVIR** : Toyohashi U. of Technology in Japan.
- **URJCyUNED** : King Juan Carlos U. in Spain.
- **MICC** : Florence U. in Italy.
- **SZTAKI** : Hungarian Academy of Sciences.
- **INAOE** : National Institute of Astrophysics, Optics and Electronics in Mexico.
- **THSSMPAM** : Tsinghua U., Beijing, China.
- **LMCHFUT** : Hefei University of Technology, China.

Performance and Rank : (ImageCLEF 2013)



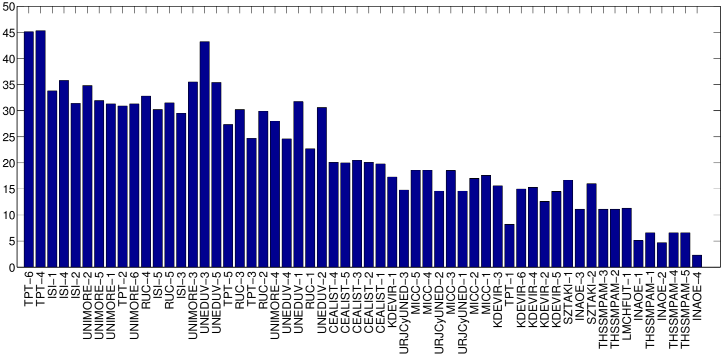
Mean F scores (samples) (see http ://imageclef.org/2013/photo/annotation/results).

Performance and Rank : (ImageCLEF 2013)



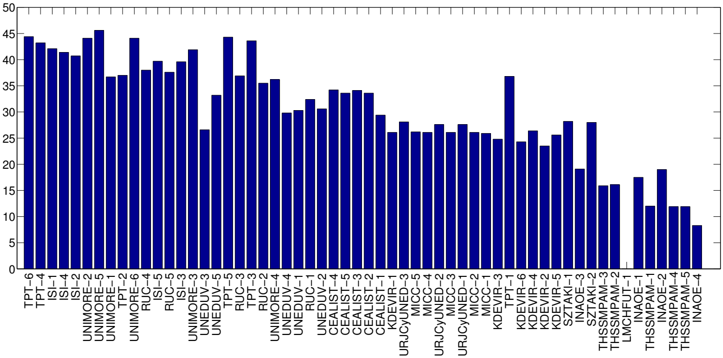
Mean F scores (concepts).

Performance and Rank : (ImageCLEF 2013)



Mean F scores (out concepts).

Performance and Rank : (ImageCLEF 2013)



MAP (concepts).

Examples (I)

GT :	cloudless moon nighttime plant silhouette sky sunrise/sunset tree
Run1 ($\tau = 0.5$) :	airplane beach boat bridge cityscape cloud cloudless coast fish fog highway lake lightning monument moon nighttime outdoor rainbow reflection sand sea silhouette sky smoke snow sun sunrise/sunset underwater water
Run2 ($\tau = 1.0$) :	beach cityscape cloudless coast lightning nighttime outdoor rainbow reflection sea silhouette sky sun sunrise/sunset water
Run3 ($\tau = 0.5$) :	beach boat bridge cityscape cloud cloudless coast fog highway lake lightning nighttime outdoor rainbow reflection sand sea silhouette sky smoke sun sunrise/sunset water
Run4 ($\tau = 1.0$) :	beach cityscape cloudless coast nighttime outdoor rainbow reflection sea silhouette sky sun sunrise/sunset water
Run5 ($\tau = 0.5$) :	boat cityscape cloud cloudless coast fog harbor lake monument nighttime outdoor reflection river sclupture sea silhouette sky smoke sun sunrise/sunset water
Run6 ($\tau = 1.0$) :	boat cityscape cloudless nighttime outdoor reflection sea silhouette sky sun sunrise/sunset water

Examples (II)

GT :	countryside daytime forest grass outdoor plant shadow tree
Run1 ($\tau = 0.5$) :	aerial bird bridge building castle countryside daytime flower forest garden grass highway horse mountain outdoor park plant rain river road shadow sky soil sport tree unpaved
Run2 ($\tau = 1.0$) :	countryside daytime forest garden grass outdoor park plant road sky soil tree unpaved
Run3 ($\tau = 0.5$) :	bicycle bridge castle countryside daytime flower forest garden grass horse mountain outdoor park plant river road shadow sky soil sport tree unpaved
Run4 ($\tau = 1.0$) :	aerial countryside daytime forest garden grass outdoor park plant sky soil sport tree unpaved
Run5 ($\tau = 0.5$) :	aerial bicycle bird countryside daytime forest grass monument motocycle mountain outdoor overcast park plant river sculpture sky soil sport tree truck unpaved
Run6 ($\tau = 1.0$) :	aerial countryside daytime forest grass outdoor park plant sky soil tree

Concepts

aerial 1 airplane 2 baby 3 beach 4 bicycle 5 bird 6 boat 7 book 8 bridge 9 building 10 car 11 cartoon 12 castle 13 cat 14 child 15 church 16 cityscape 17 closeup 18 cloud 19 cloudless 20 coast 21 countryside 22 daytime 23 desert 24 diagram 25 dog 26 drum 27 elder 28 embroidery 29 female 30 fire 31 firework 32 fish 33 flower 34 fog 35 food 36 footwear 37 forest 38 furniture 39 garden 40 grass 41 guitar 42 harbor 43 helicopter 44 highway 45 horse 46 indoor 47 instrument 48 lake 49 lightning 50 logo 51 male 52 monument 53 moon 54 motorcycle 55 mountain 56 newspaper 57 nighttime 58 outdoor 59 overcast 60 painting 61 paint 61 park 62 person 63 plant 64 portrait 65 poster 66 protest 67 rain 68 rainbow 69 reflection 70 river 71 road 72 sand 73 sculpture 74 sea 75 shadow 76 sign 77 silhouette 78 sky 79 smoke 80 snow 81 soil 82 sport 83 sun 84 sunrise/sunset 85 teenager 86 toy 87 traffic 88 train 89 tree 90 truck 91 underwater 92 unpaved 93 vehicle 94 water 95 . . .

Outline

- Introduction
- Kernels (on non vectorial data)
- Context in kernel design
- Context dependent image annotation
- Conclusion and take home message

Conclusion

- **Take Home Message**
  - Similarity in kernel design is not only intrinsic, it also depends on the context.
  - Context dependent kernels show better annotation results with respect to context free kernels.
  - The positive definiteness of CDK makes it possible to **map non vectorial data** (image+links) explicitly into vectorial ones.
  - **Convergence** properties are valid.

Conclusion

- **Take Home Message**
  - Similarity in kernel design is not only intrinsic, it also depends on the context.
  - Context dependent kernels show better annotation results with respect to context free kernels.
  - The positive definiteness of CDK makes it possible to **map non vectorial data** (image+links) explicitly into vectorial ones.
  - **Convergence** properties are valid.

Conclusion

- **Take Home Message**
  - Similarity in kernel design is not only intrinsic, it also depends on the context.
  - Context dependent kernels show better annotation results with respect to context free kernels.
  - The positive definiteness of CDK makes it possible to **map non vectorial data** (image+links) explicitly into vectorial ones.
  - **Convergence** properties are valid.

Conclusion

- **Take Home Message**
  - Similarity in kernel design is not only intrinsic, it also depends on the context.
  - Context dependent kernels show better annotation results with respect to context free kernels.
  - The positive definiteness of CDK makes it possible to **map non vectorial data** (image+links) explicitly into vectorial ones.
  - **Convergence** properties are valid.